#3 Priority Paper MAH 5/13/02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Attorney Docket No. 067238/0115

Applicant: Atsufumi SHIBAYAMA, et al.

Title: DATA PROCESSOR WITH AN IMPROVED DATA DEPENDENCE DETECTOR

Serial No.: 09/994,806

Filed: November 28, 2001

Examiner: Unassigned

Art Unit: 2152

CLAIM FOR CONVENTION PRIORITY

Commissioner for Patents
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested, and the right of priority provided in 35 U.S.C. § 119 is hereby claimed.

In support of this claim, filed herewith is a certified copy of said original foreign application:

Japan Patent Application No. 2000-363727 filed 11/29/2000.

Respectfully submitted,

JANUARY 18, 2002

Date

Ankur D. Shah
Registration No. 41,514

FOLEY & LARDNER
Washington Harbour
3000 K Street, N.W. – Suite 500
Washington, D.C. 20007
Telephone: (202) 672-5300
Facsimile: (202) 672-5399

US

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２０００年１１月２９日

出願番号
Application Number: 特願２０００－３６３７２７

出願人
Applicant(s): 日本電気株式会社

２００１年　８月３１日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

出証番号　出証特２００１－３０７７８１３

【書類名】　特許願
【整理番号】　34002109
【提出日】　平成12年11月29日
【あて先】　特許庁長官　殿
【国際特許分類】　G06F　9/46
　　G06F 15/16
【発明者】
　【住所又は居所】　東京都港区芝五丁目７番１号　日本電気株式会社内
　【氏名】　柴山　充文
【発明者】
　【住所又は居所】　東京都港区芝五丁目７番１号　日本電気株式会社内
　【氏名】　松下　智
【発明者】
　【住所又は居所】　東京都港区芝五丁目７番１号　日本電気株式会社内
　【氏名】　鳥居　淳
【発明者】
　【住所又は居所】　東京都港区芝五丁目７番１号　日本電気株式会社内
　【氏名】　西　直樹
【特許出願人】
　【識別番号】　000004237
　【氏名又は名称】　日本電気株式会社
【代理人】
　【識別番号】　100080816
　【弁理士】
　【氏名又は名称】　加藤　朝道
　【電話番号】　045-476-1131
【手数料の表示】
　【予納台帳番号】　030362
　【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9304371

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　データ依存関係検出装置

【特許請求の範囲】

【請求項１】

非プログラム順序でメモリ操作命令を実行するに際して、前記メモリ操作命令の処理アドレス間の依存関係の有無を検出するデータ依存関係検出装置において、

前記メモリ操作命令の処理アドレス間の依存関係が存在しない場合にも依存関係が存在する旨を検出する場合があることを許容する構成とされている、ことを特徴とする、データ依存関係検出装置。

【請求項２】

非プログラム順序でメモリ操作命令を実行するに際して、前記メモリ操作命令の処理アドレス間の依存関係の有無を検出するデータ依存関係検出装置において、

前記メモリ操作命令の処理アドレス間の依存関係が存在する場合には必ず依存関係が存在する旨を検出すると共に、前記依存関係が存在しない場合にも依存関係が存在する旨を検出する場合があることを許容する構成とされている、ことを特徴とする、データ依存関係検出装置。

【請求項３】

非プログラム順序でメモリ操作命令を実行するに際して、前記メモリ操作命令の処理アドレス間の依存関係の有無を検出するデータ依存関係検出装置において、

前記メモリ操作命令の処理アドレスを受け該処理アドレスを一意な番号に変換するアドレス変換手段と、

前記アドレス変換手段で変換された番号に対応して、メモリ操作命令が実行されたか否かを示す値を記憶する複数の記憶手段を有する実行履歴記憶手段と、

を少なくとも含み、

前記メモリ操作命令の処理アドレス間の依存関係が存在する場合には必ず依存関係が存在する旨を検出すると共に、前記依存関係が存在しない場合にも依存関

係が存在する旨を検出する場合がある、ことを許容する、ことを特徴とする、データ依存関係検出装置。

【請求項４】

請求項３記載のデータ依存関係検出装置において、

前記メモリ操作命令のロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、該ロード命令が実行されたことを示す値を格納し、

前記メモリ操作命令のストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されているロード命令が実行されたか否かを示す値を読み出すことで、前記実行されたストア命令から、前記実行されたロード命令への正依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項５】

請求項３記載のデータ依存関係検出装置において、

前記メモリ操作命令のストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、該ストア命令が実行されたことを示す値を格納し、

前記メモリ操作命令のロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されているストア命令が実行されたか否かを示す値を読み出すことで、前記実行されたロード命令から、前記実行されたストア命令への逆依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項６】

請求項３記載のデータ依存関係検出装置において、

前記メモリ操作命令のストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、ストア命令が実行されたか否かを示す値を読み出すことに加えて、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、該ストア命令が実行されたことを示す値を格納することで、前記実行されたストア命令間に出力依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項７】

請求項３記載のデータ依存関係検出装置において、

前記メモリ操作命令のロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、前記メモリ操作命令のストア命令が実行されたか否かを示す値を読み出すことに加えて、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段にロード命令が実行されたことを示す値を格納し、

ストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されているロード命令が実行されたか否かを示す値及びストア命令が実行されたか否かを示す値を読み出すことに加えて、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段にストア命令が実行されたことを示す値を格納し、前記実行されたストア命令から、前記実行されたロード命令への正依存関係が存在する可能性、及び前記実行されたロード命令から、前記実行されたストア命令への逆依存関係が存在する可能性、及び前記実行されたストア命令間の出力依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項８】

前記アドレス変換手段は、入力されるアドレスの所定のビットをそのまま出力することで、任意のアドレスを前記所定のビットで規定される番号に変換する、構成とされている、ことを特徴とする、請求項３から請求項７のいずれか一に記載のデータ依存関係検出装置。

【請求項９】

前記アドレス変換手段は、入力されるアドレスの複数の所定のビットの排他的論理和を出力する手段を備え、任意のアドレスを前記排他的論理和出力で規定される番号に変換する、構成とされている、ことを特徴とする、請求項３から請求項７のいずれか一に記載のデータ依存関係検出装置。

【請求項１０】

複数のプロセッサから構成されるマルチプロセッサシステムにおける前記各々のプロセッサが備えるデータ依存関係検出装置であって、前記マルチプロセッサシステムがスレッド単位で並列処理を行う際に、他のプロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスと、自プロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスとの間の依存関係の有無を検出するデータ依存関係検出装置において、

依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容する構成とされている、ことを特徴とする、データ依存関係検出装置。

【請求項１１】

複数のプロセッサから構成されるマルチプロセッサシステムにおける前記各々のプロセッサが備えるデータ依存関係検出装置であって、前記マルチプロセッサシステムがスレッド単位で並列処理を行う際に、他のプロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスと、自プロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスとの間の依存関係の有無を検出するデータ依存関係検出装置において、

前記依存関係が存在する場合は必ず依存関係が存在する旨を検出すると共に、前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容する構成とされている、ことを特徴とする、データ依存関係検出装置

【請求項１２】

複数のプロセッサから構成されるマルチプロセッサシステムにおける前記各々のプロセッサが備えるデータ依存関係検出装置であって、前記マルチプロセッサシステムがスレッド単位で並列処理を行う際に、他のプロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスと、自プロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスとの間の依存関係の有無を検出するデータ依存関係検出装置において、

自プロセッサ及び他プロセッサが実行するメモリ操作命令の処理アドレスを一意な番号に変換する複数のアドレス変換手段と、

前記変換された番号に対応して、メモリ操作命令が実行されたか否かを記憶する複数の記憶手段から構成される実行履歴記憶手段と、

を少なくとも含み、前記依存関係が存在する場合は必ず依存関係が存在する旨を検出すると共に、前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容する、ことを特徴とする、データ依存関係検出装置。

【請求項１３】

請求項１２記載のデータ依存関係検出装置において、

自プロセッサでロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、自プロセッサでロード命令が実行されたことを示す値を格納し、他のプロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでロード命令が実行されたか否かを示す値を読み出すことで、前記他のプロセッサで実行されたストア命令から、前記自プロセッサで実行されたロード命令への正依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項１４】

請求項１２記載のデータ依存関係検出装置において、

自プロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、自プロセッサでストア命令が実行されたことを示す値を格納し、他のプロセッサでロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでストア命令が実行されたか否かを示す値を読み出すことで、前記他のプロセッサで実行されたロード命令から、前記自プロセッサで実行されたストア命令への逆依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項１５】

請求項１２記載のデータ依存関係検出装置において、

他のプロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでストア命令が実行されたか否かを示す値を読み出し、自プロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、自プロセッサでストア命令が実行されたことを示す値を格納することで、前記他のプロセッサで実行されたストア命令から、前記自プロセッサで実行されたストア命令への出力依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項１６】

請求項１２記載のデータ依存関係検出装置において、

他のプロセッサでロード命令が実行された場合には、該ロード命令が処理対象

とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでストア命令が実行されたか否かを示す値を読み出し、他のプロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでロード命令が実行されたか否かを示す値及び自プロセッサでストア命令が実行されたか否かを示す値を読み出し、自プロセッサでロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、自プロセッサでロード命令が実行されたことを示す値を格納し、自プロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、自プロセッサでストア命令が実行されたことを示す値を格納することで、前記他プロセッサで実行されたストア命令から、前記自プロセッサで実行されたロード命令への正依存関係が存在する可能性、及び前記他プロセッサで実行されたロード命令から、前記自プロセッサで実行されたストア命令への逆依存関係が存在する可能性、及び前記他のプロセッサで実行されたストア命令から、前記自プロセッサで実行されたストア命令への出力依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項１７】

前記アドレス変換手段が、入力されるアドレスの所定のビットをそのまま出力することで、任意のアドレスを前記所定のビットで規定される番号に変換する、構成とされている、ことを特徴とする、請求項１２から請求項１６のいずれか一に記載のデータ依存関係検出装置。

【請求項１８】

前記アドレス変換手段が、入力されるアドレスの複数の所定のビットの排他的

論理和を出力する手段を備え、任意のアドレスを前記排他的論理和出力で規定される番号に変換する、構成とされている、ことを特徴とする、請求項１２から請求項１６のいずれか一に記載のデータ依存関係検出装置。

【請求項１９】

請求項１３から請求項１６のいずれか一に記載のデータ依存関係検出装置において、

他のプロセッサのうち、自プロセッサが実行するスレッドよりもプログラム順序で前に位置するスレッドを実行するプロセッサにおいてメモリ操作命令が実行された場合にのみ、該メモリ操作命令の処理アドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されているメモリ操作命令が実行されたか否かを示す値を読み出す、ことを特徴とする、データ依存関係検出装置。

【請求項２０】

請求項１３から請求項１６のいずれか一に記載のデータ依存関係検出装置において、

他のプロセッサがメモリ操作命令を実行する場合に、該メモリ操作命令の処理アドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されているメモリ操作命令が実行されたか否かを示す値を読み出す際に、自プロセッサが実行するスレッドよりもプログラム順序で前に位置するスレッドを実行するプロセッサが実行したメモリ操作命令により読み出された値のみを参照する、ことを特徴とする、データ依存関係検出装置。

【請求項２１】

複数のプロセッサから構成されるマルチプロセッサシステムにおいて、前記各々のプロセッサが備えるデータ依存関係検出装置であって、かつ前記マルチプロセッサシステムがスレッド単位で並列処理を行う際に、他のプロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスと、自プロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスとの間の依存関係の有無、及び自プ

ロセッサが実行するスレッドが含むメモリ操作命令の対象アドレス間の依存関係を検出するデータ依存関係検出装置において、

前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容する構成とされている、ことを特徴とする、データ依存関係検出装置。

【請求項２２】

複数のプロセッサから構成されるマルチプロセッサシステムにおいて、前記各々のプロセッサが備えるデータ依存関係検出装置であって、かつ前記マルチプロセッサシステムがスレッド単位で並列処理を行う際に、他のプロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスと、自プロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスとの間の依存関係の有無、及び自プロセッサが実行するスレッドが含むメモリ操作命令の対象アドレス間の依存関係を検出するデータ依存関係検出装置において、

前記依存関係が存在する場合は必ず依存関係が存在する旨を検出すると共に、前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容する、構成とされている、ことを特徴とする、データ依存関係検出装置。

【請求項２３】

複数のプロセッサから構成されるマルチプロセッサシステムにおいて、前記各々のプロセッサが備えるデータ依存関係検出装置であって、かつ前記マルチプロセッサシステムがスレッド単位で並列処理を行う際に、他のプロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスと、自プロセッサが実行するスレッドが含むメモリ操作命令の処理アドレスとの間の依存関係の有無、及び自プロセッサが実行するスレッドが含むメモリ操作命令の対象アドレス間の依存関係を検出するデータ依存関係検出装置において、

自プロセッサ及び他プロセッサが実行するメモリ操作命令の処理アドレスを一意な番号に変換する複数のアドレス変換手段と、

前記変換された番号に対応して、メモリ操作命令が実行されたか否かを記憶する複数の記憶手段から構成される実行履歴記憶手段と、を少なくとも含み、

前記依存関係が存在する場合は必ず依存関係が存在する旨を検出すると共に、前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容する、ことを特徴とする、データ依存関係検出装置。

【請求項２４】

請求項２３記載のデータ依存関係検出装置において、

自プロセッサでロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、自プロセッサでロード命令が実行されたことを示す値を格納し、

自プロセッサあるいは他のプロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでロード命令が実行されたか否かを示す値を読み出すことで、前記自プロセッサあるいは他のプロセッサで実行されたストア命令から、前記自プロセッサで実行されたロード命令への正依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項２５】

請求項２３記載のデータ依存関係検出装置において、

自プロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、自プロセッサでストア命令が実行されたことを示す値を格納し、

自プロセッサあるいは他のプロセッサでロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでストア命令が実行されたか否かを示す値を読み出すことで、前記自プロセッサあるいは他のプロセッサで実行されたロード命令から、前記自プロセッサで実行されたストア命令

への逆依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項２６】

請求項２３記載のデータ依存関係検出装置において、

他のプロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている自プロセッサでストア命令を実行したか否かを示す値を読み出し、

自プロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでストア命令を実行したか否かを示す値を読み出すのに加えて、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に、自プロセッサでストア命令が実行されたことを示す値を格納することで、前記自プロセッサあるいは他のプロセッサで実行されたストア命令から、前記自プロセッサで実行されたストア命令への出力依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項２７】

請求項２３記載のデータ依存関係検出装置において、

自プロセッサあるいは他のプロセッサでロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されている、自プロセッサでストア命令が実行されたか否かを示す値を読み出し、

自プロセッサあるいは他のプロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換さ

れた番号に対応する記憶手段に格納されている、自プロセッサでロード命令が実行されたか否かを示す値及び自プロセッサでストア命令が実行されたか否かを示す値を読み出すことに加えて、自プロセッサでロード命令が実行された場合には、該ロード命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に自プロセッサでロード命令が実行されたことを示す値を格納し、

自プロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に自プロセッサでストア命令が実行されたことを示す値を格納することで、前記自プロセッサあるいは他プロセッサで実行されたストア命令から、前記自プロセッサで実行されたロード命令への正依存関係が存在する可能性、及び、前記自プロセッサあるいは他プロセッサで実行されたロード命令から、前記自プロセッサで実行されたストア命令への逆依存関係が存在する可能性、及び前記自プロセッサあるいは他のプロセッサで実行されたストア命令から、前記自プロセッサで実行されたストア命令への出力依存関係が存在する可能性を検出する、ことを特徴とする、データ依存関係検出装置。

【請求項２８】

前記アドレス変換手段が、入力されるアドレスの所定のビットをそのまま出力することで、任意のアドレスを前記所定のビットで規定される番号に変換する、構成とされている、ことを特徴とする、請求項２３から請求項２７のいずれか一に記載のデータ依存関係検出装置。

【請求項２９】

前記アドレス変換手段が、入力されるアドレスの複数の所定位置のビットの排他的論理和を出力する手段を備え、任意のアドレスを前記排他的論理和出力で規定される番号に変換する、構成とされている、ことを特徴とする、請求項２３から請求項２７のいずれか一に記載のデータ依存関係検出装置。

【請求項３０】

請求項２４から請求項２７のいずれか一に記載のデータ依存関係検出装置において、

他のプロセッサのうち、自プロセッサが実行するスレッドよりもプログラム順序で前に位置するスレッドを実行するプロセッサにおいてメモリ操作命令が実行された場合にのみ、該メモリ操作命令の処理アドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されているメモリ操作命令が実行されたか否かを示す値を読み出す、ことを特徴とする、データ依存関係検出装置。

【請求項３１】

請求項２４から請求項２７のいずれか一に記載のデータ依存関係検出装置において、

他のプロセッサがロード命令またはストア命令を実行する場合に、該ロード命令または該ストア命令が処理対象とするアドレスを前記アドレス変換手段により一意な番号に変換し、前記実行履歴記憶手段を構成する複数の記憶手段のうち、前記変換された番号に対応する記憶手段に格納されているロード命令またはストア命令が実行されたか否かを示す値を読み出す際に、自プロセッサが実行するスレッドよりもプログラム順序で前に位置するスレッドを実行するプロセッサが実行したロード命令またはストア命令により読み出された値のみを参照する、ことを特徴とする、データ依存関係検出装置。

【請求項３２】

請求項１から請求項３１のいずれか一に記載のデータ依存関係検出装置を搭載した半導体集積回路装置。

【請求項３３】

前記メモリ操作命令の処理アドレス間の依存関係が存在しない場合にも依存関係が存在する旨を検出する場合があることを許容する構成が、複数のエントリよりなり、前記メモリ操作命令の命令履歴を管理する実行履歴記憶手段へのエントリ番号を前記メモリ操作命令のアドレスから生成するアドレス変換手段を含み、

前記アドレス変換回路は、前記メモリ操作命令の処理対象アドレスを、前記実

行履歴記憶手段の総エントリ数の範囲内のエントリ番号に変換し、同一の対象アドレスに対しては同一のエントリ番号を出力するとともに、同一のエントリ番号に対して異なる対象アドレスが重複して存在することが許容される構成とされ、前記メモリ操作命令の処理アドレス間の依存関係が存在しない場合にも、依存関係が存在する旨を検出するものである、ことを特徴とする、請求項１、２、１１、１２、２１、２２のいずれか一に記載のデータ依存関係検出装置。

【請求項３４】

前記アドレス変換手段が、複数のエントリよりなり、前記メモリ操作命令の命令履歴を管理する実行履歴記憶手段へのエントリ番号を、前記メモリ操作命令のアドレスから生成するにあたり、前記メモリ操作命令の処理対象アドレスを、前記実行履歴記憶手段の総エントリ数の範囲内のエントリ番号に変換し、同一の対象アドレスに対しては同一のエントリ番号を出力するとともに、同一のエントリ番号に対して異なる対象アドレスが重複して存在することが許容される構成とされている、ことを特徴とする、請求項３乃至９、請求項１２乃至２０、請求項２３乃至３１のいずれか一に記載のデータ依存関係検出装置。

【請求項３５】

命令実行部と、プロセッサ制御部と、データ依存関係検出装置とを少なくとも含むプロセッサ装置において、

前記プロセッサ制御部は、前記プロセッサ装置が命令を実行するにあたり、実行する命令の種別を前記命令実行部、及び、前記データ依存関係検出装置に出力し、その際、実行する命令が、メモリから汎用レジスタへのデータの転送を行うロード命令、または、汎用レジスタからメモリへのデータ転送を行うストア命令よりなるメモリ操作命令である場合には、該メモリ操作命令の処理対象アドレスを、ロード・ストアアドレス線から、前記命令実行部、及び、前記データ依存関係検出装置に出力するとともに、ロード命令またはストア命令をデータ依存投機実行する場合に、その旨を、投機実行フラグを通じて、前記データ依存関係検出装置に通知する構成とされ、

前記データ依存関係検出装置は、前記プロセッサ制御部から出力されるロード命令とストア命令の処理対象アドレスを、前記ロード・ストアアドレス線から入

力し、前記対象アドレスを、前記対象アドレスのビット幅よりも小のビット幅で表現されるエントリ番号に変換し、ロード命令とストア命令について同一の対象アドレスに対しては同一のエントリ番号を出力し、異なる対象アドレスに対して同一のエントリ番号が出力され、同一のエントリ番号に対して異なる対象アドレスが存在する場合がある構成よりなるハッシュ関数回路と、

複数のエントリから構成され、前記ハッシュ関数回路が出力するエントリ番号が指し示すエントリに対して、前記プロセッサ制御部が出力する前記実行命令と前記投機実行フラグとを参照して、該エントリの書き込みや読み出し処理を行う命令履歴テーブルと、

を備え、データ依存投機実行が失敗したか成功したかを示す値をデータ依存検出結果に出力して前記プロセッサ制御部に通知する、ことを特徴とする、プロセッサ装置。

【請求項３６】

前記プロセッサ装置がデータ依存関係に対して投機的な命令実行を行う状態にある場合、投機的なロード命令が実行されると、前記データ依存関係検出装置は、投機的なロード命令の対象アドレスを、前記ハッシュ関数回路によりエントリ番号に変換して前記命令履歴テーブルに入力し、

前記命令履歴テーブルは、入力されたエントリ番号に対応するエントリを、投機的なロード命令が実行されたことを示す値に設定し、

投機的なストア命令が実行された場合には、ストア命令の対象アドレスを前記ハッシュ関数回路によりエントリ番号に変換して前記命令履歴テーブルに入力し、

前記命令履歴テーブルは、入力されたエントリ番号に対応するエントリに格納されている値を読み出し、前記読み出した値に基づいて、ストア命令から実行されたロード命令へ正依存関係が存在する可能性があるか否かを検出し、正依存関係が存在する可能性があることが検出された場合、前記データ依存関係検出装置は、データ依存投機実行が失敗したことを示す値をデータ依存検出結果に出力する、ことを特徴とする、請求項３５に記載のプロセッサ装置。

【請求項３７】

前記データ依存関係検出装置は、前記プロセッサ制御部から前記データ依存関係検出装置に入力される前記投機実行フラグに基づいて、前記命令履歴テーブルのすべてのエントリに同一の値を書き込む初期化機能を有する、ことを特徴とする、請求項３５又は３６に記載のプロセッサ装置。

【請求項３８】

前記データ依存関係検出装置からデータ依存投機実行が失敗したことを示す値が前記データ依存検出結果に出力された場合、前記プロセッサ制御部及び前記命令実行部は、データ依存投機実行が失敗したことに対する回復処理を実行する、ことを特徴とする、請求項３５又は３６に記載のプロセッサ装置。

【請求項３９】

前記データ依存関係検出装置は、ストア命令からロード命令へ、正依存関係が存在しないことが検出された場合、データ依存投機実行が成功したことを示す値を前記データ依存検出結果に出力し、これを受けて前記プロセッサ制御部はデータ依存投機実行が成功したとして、データ依存投機実行が失敗したことに対する回復処理は実行せずに、後続の命令実行を続行する、ことを特徴とする、請求項３５に記載のプロセッサ装置。

【請求項４０】

プロセッサ制御部、データ依存関係検出装置、及び命令実行部を含むプロセッサを複数（Ｎ個）備え、

プログラムをプロセッサによってスレッド並列処理するにあたり、前記複数のプロセッサの各々が実行すべきスレッドの割り当てを行い、前記複数のプロセッサのそれぞれに、スレッド実行がデータ依存投機実行か否かを示す投機実行フラグを出力し、さらに一のプロセッサと他のプロセッサが実行するスレッド間の順序関係を示すスレッド順序を出力するスレッド制御部と、

前記各プロセッサは、それぞれ、投機実行フラグ及びスレッド順序を入力し、スレッド制御部より割り当てられたスレッドを、各プロセッサが実行する場合、各プロセッサの前記プロセッサ制御部は、自プロセッサが実行する命令の種別を、実行命令線から自プロセッサの前記命令実行部と前記データ依存関係検出装置とに出力するとともに、実行命令線を通じて他のプロセッサに出力し、

前記データ依存関係検出装置は、自プロセッサが実行する命令種別を実行命令線から入力するとともに、他のプロセッサが実行する命令種別を、前記他のプロセッサの前記プロセッサ制御部から出力される実行命令線を通じてそれぞれ入力し、

実行する命令がロード命令またはストア命令の場合、前記プロセッサ制御部は、ロード命令またはストア命令の対象アドレスを、ロード・ストアアドレス線から、自プロセッサの前記命令実行部と前記データ依存関係検出装置に出力するとともに、ロード・ストアアドレス線を通じて他プロセッサに出力し、

前記データ依存関係検出装置は、プロセッサ個数分（Ｎ個）のハッシュ関数回路と、命令履歴テーブルと、論理和回路とを備え、

自プロセッサの前記プロセッサ制御部からの入力として、実行命令線、ロード・ストアアドレス線を入力するとともに、及び、他のプロセッサの前記プロセッサ制御部から、それぞれの実行命令線とロード・ストアアドレス線とを入力とし、

前記スレッド制御部からは投機実行フラグを入力し、前記投機実行フラグに基づき、自プロセッサが確定実行状態または投機実行状態のどちらにあるかを判断し、

自プロセッサが実行するスレッドが、他のプロセッサが実行するそれぞれのスレッドよりもプログラム順序で前の順序にあるか否かは、前記スレッド制御部から前記複数のプロセッサの前記データ依存関係検出装置にそれぞれ入力されるスレッド順序で判断し、さらに、前記スレッド制御部へは、データ依存検出結果を出力し、

前記ハッシュ関数回路は、ロード命令またはストア命令の対象アドレスを、命令履歴テーブルの総エントリ数の範囲内のエントリ番号に変換するハッシュ関数を実現する論理回路からなり、同一の入力に対しては同一の値をもち、前記ハッシュ関数回路のうちの一つは、自プロセッサが実行するロード命令の対象アドレスを命令履歴テーブルのエントリ番号に変換し、残りのＮ－１個の前記ハッシュ関数回路はそれぞれ、対応する他のプロセッサが実行するストア命令の対象アドレスを対応する命令履歴テーブのエントリ番号に変換し、

前記命令履歴テーブルは、複数のエントリから構成され、一の書き込みポートとＮ－１個の読み出しポートとを備え、

自プロセッサが実行するロード命令の対象アドレスを入力とする前記ハッシュ関数回路の出力は、前記命令履歴テーブルの書き込みポートに接続され、前記ハッシュ関数回路が出力するエントリ番号が指し示すエントリに対して書き込み処理を行い、

他のプロセッサが実行するストア命令の対象アドレスを入力とする前記ハッシュ関数回路の出力は、対応する命令履歴テーブルの読み出しポートに接続され、前記ハッシュ関数回路が出力するエントリ番号が指し示すエントリに対して読み出し処理を行い、

前記論理和回路は、前記命令履歴テーブルが備えるＮ－１個の読み出しポートのＮ－１個の読み出し結果に対して論理和を演算した結果をデータ依存検出結果として前記スレッド制御部に出力する、ことを特徴とする、マルチプロセッサ装置。

【請求項４１】

前記スレッド制御部から前記データ依存関係検出装置に入力される前記投機実行フラグに基づいて前記命令履歴テーブルのすべてのエントリに同一の値を書き込む初期化機能を有する、ことを特徴とする、請求項４０記載のマルチプロセッサ装置。

【請求項４２】

自プロセッサが投機実行状態にあるときに、投機的なロード命令が実行されると、前記データ依存関係検出装置は、ロード命令の対象アドレスを、対応する一の前記ハッシュ関数回路により前記命令履歴テーブルのエントリ番号に変換して前記命令履歴テーブルの書き込みポートに入力し、前記命令履歴テーブルは、入力されたエントリ番号に対応するエントリを、投機的なロード命令が実行されたことを示す値に設定し、

自プロセッサが投機実行状態にあるときに、他のプロセッサのうち、自プロセッサが実行するスレッドよりもプログラム順序で前の順序にあるスレッド、すなわち先行するスレッドを割り当てられたプロセッサでストア命令が実行された場

合には、ストア命令の対象アドレスを前記ハッシュ関数回路により前記命令履歴テーブルのエントリ番号に変換し、前記命令履歴テーブルのＮ－１個の入力ポートのうち対応するものに入力し、前記命令履歴テーブルは、入力されたエントリ番号に対応するエントリの内容を読み出して前記論理和回路に出力し、

前記論理和回路は、前記命令履歴テーブルのＮ－１の読み出しポートの出力の論理和を演算し前記データ依存検出結果として前記スレッド制御部に出力する、ことを特徴とする、請求項４０又は４１に記載のマルチプロセッサ装置。

【請求項４３】

前記複数（Ｎ個）のプロセッサにおいて、一のプロセッサは、他のプロセッサのうち一のプロセッサが実行するスレッドよりも先行するスレッドを実行するプロセッサのいずれかが実行したストア命令の対象アドレスが、前記一のプロセッサがそれまでデータ依存投機実行したロード命令の対象アドレスと等しいか、あるいは等しくない場合でも、前記命令履歴テーブルの同じエントリに割り当てられた場合、前記命令履歴テーブルからデータ依存投機実行されたロード命令が存在することを示す値が読み出され、正依存関係が存在する可能性があることが検出され、前記一のプロセッサの前記データ依存関係検出装置は、データ依存投機実行が失敗したことを示す値を、前記データ依存検出結果を通じて、前記スレッド制御部に出力し、

前記スレッド制御部は、Ｎ個のプロセッサのいずれかより、データ依存投機実行の失敗の通知を受け取ると、その失敗通知を出力したプロセッサ、及び、その失敗通知を出力したプロセッサが実行していたスレッドよりも、プログラム順序で後に位置するスレッドを実行しているプロセッサに対して、データ依存投機実行の失敗による回復処理の実行要求を出力し、

データ依存投機実行の失敗による回復処理の実行要求の出力対象となった各プロセッサでは、データ依存投機実行の失敗による回復処理の実行要求が、回復処理実行要求信号を通じて、前記プロセッサ制御部に通知される、ことを特徴とする、請求項４０又は４１に記載のマルチプロセッサ装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、マイクロプロセッサ等のデータ処理装置に関し、特にマイクロプロセッサの性能向上技術に関する。

【０００２】

【従来の技術】

従来より、マイクロプロセッサの性能向上技術として、マイクロプロセッサが実行するプログラム中の命令の順序とは異なる順序で命令の実行を行う非プログラム順序実行（アウトオブオーダ実行）技術が広く使われている。プロセッサが非プログラム順序実行を行う場合、実行可能になった命令から実行を行うため、プログラム中の命令の順序で実行を行うプログラム順序実行（インオーダ実行）を行うのに比べて、プロセッサの性能を向上させることが可能となる。

【０００３】

しかしながら、非プログラム順序実行は無条件に行えるわけではない。非プログラム順序実行が可能な条件として、まず、実行する命令間のレジスタに関する正依存関係がないことがあげられる。

【０００４】

プログラム順序で前にある命令が変更を行うレジスタを、プログラム順序でそれより後にある命令が参照する場合、前にある命令から後にある命令へレジスタに関する正依存関係があるという。この場合、それら２つの命令をプログラム順序とは逆の順序で実行を行うと、プログラムの意味が変わってしまい、正しいプログラムの実行結果を得ることができない。すなわち、レジスタに関して正依存関係がある場合、非プログラム順序で実行を行うことができない。

【０００５】

同様にして、プログラム順序で前にある命令が参照するレジスタを、プログラム順序でそれより後にある命令が変更を行う場合、前にある命令から後にある命令へレジスタに関する逆依存関係があるという。

【０００６】

また、プログラム順序で前にある命令が変更を行うレジスタを、プログラム順序でそれより後にある命令も変更を行う場合、前にある命令から後にある命令へ

レジスタに関する出力依存関係があるという。

【０００７】

いずれの場合も、それら２つの命令をプログラム順序とは逆の順序で実行を行うと、プログラムの意味が変わってしまい、正しいプログラムの実行結果を得ることができない。

【０００８】

通常、各命令が変更あるいは参照の対象とするレジスタは、命令のデコード時に判明するため、正依存関係のある命令に対しては、非プログラム順序で実行を行わないことで対応することが多い。

【０００９】

また、逆依存関係、出力依存関係に対しては、レジスタリネーミング等の技術で依存を解消して、非プログラム順序実行を可能にすることも行われている。

【００１０】

メモリに対して操作を行う命令（「メモリ操作命令」という）に対しては、非プログラム順序実行に際して、レジスタに関する依存関係に加えて、メモリに関する依存関係に対して考慮が必要となる。

【００１１】

通常、メモリ操作命令には、メモリからデータを読み出すロード命令と、メモリへデータを書き込むストア命令が含まれる。

【００１２】

プログラム中に出現する、ある２つのロード命令またはストア命令が対象とするアドレスが互いに異なれば、メモリの異なる位置が読み出しまたは書き込みの対象となるため、それら２つのロード命令またはストア命令の間にはメモリに関して依存関係は存在しない。すなわち、それら２つの命令は非プログラム順序で実行することが可能である。

【００１３】

一方、プログラム中に出現するある２つのロード命令またはストア命令が対象とするアドレスが同じ場合、メモリの同じ位置が読み出しまたは書き込みの対象となるため、それら２つのロード命令またはストア命令間には、メモリに関して

依存関係が存在する。例えば、プログラム順序で前にあるストア命令が書き込みを行うアドレスに対して、プログラム順序でそれより後にあるロード命令が読み出しを行う場合、前にあるストア命令から後にあるロード命令へメモリに関する正依存関係があるという。この場合、それら２つのロード命令、ストア命令をプログラム順序とは逆の順序で実行を行うと、プログラムの意味が変わってしまい、正しいプログラムの実行結果を得ることができない。すなわち、メモリに関して正依存関係がある場合、非プログラム順序で実行を行うことができない。

【００１４】

同様にして、プログラム順序で前にあるロード命令が読み出しを行うアドレスに対して、プログラム順序でそれより後にあるストア命令が書き込みを行う場合、前にあるロード命令から後にあるストア命令へメモリに関する逆依存関係があるという。

【００１５】

また、プログラム順序で前にあるストア命令が書き込みを行うアドレスに対して、プログラム順序でそれより後にあるストア命令も書き込みを行う場合、前にあるストア命令から後にあるストア命令へメモリに関する出力依存関係があるという。

【００１６】

いずれの場合も、それら２つのロード・ストア命令をプログラム順序とは逆の順序で実行を行うと、プログラムの意味が変わってしまい、正しいプログラムの実行結果を得ることができない。

【００１７】

メモリに関する逆依存関係及び出力依存関係に対しては、ストア命令が書き込みを行うデータを一時的なバッファ（このバッファは、通常、「ストアバッファ」と呼ばれる）に一時的に格納するなどの対応で、逆依存関係あるいは出力依存関係を解消し、非プログラム順序実行を可能とする技術が、従来より、知られている。

【００１８】

一方、正依存関係が存在する場合は、本質的に非プログラム順序で実行を行う

ことができず、プログラム順序で実行を行うことになる。

【００１９】

しかしながら、ロード・ストア命令が対象とするアドレスは実行時まで不明である場合が多い。すなわち、デコード時に依存関係が判明するレジスタの場合と違い、メモリの場合には実行時まで依存関係が判明しない場合が多いため、非プログラム順序で実行を行えないことによる性能への影響が大きい。具体例を図９を参照して説明する。

【００２０】

図９（ａ）は、ロード命令とストア命令からなるプログラム例のプログラム順序を示した図である。プログラム順に、

アドレスＡ１に対するロード命令ＬＤ１、

アドレスＡ４’に対するストア命令ＳＴ１、

アドレスＡ２に対するロード命令ＬＤ２、

アドレスＡ３に対するロード命令ＬＤ３、

アドレスＡ４に対するロード命令ＬＤ４、

を示している。

【００２１】

ここで、ＳＴ１の対象アドレスＡ４’と、ＬＤ４の対象アドレスＡ４が等しい（Ａ４’＝Ａ４）とすると、ＳＴ１とＬＤ４は同じアドレスを対象としており、かつ、ＬＤ４よりもＳＴ１の方がプログラム順序で前にあるため、ＳＴ１からＬＤ４へ正依存関係が存在する。

【００２２】

つまり、プログラムの意味上、ＳＴ１がアドレスＡ４’（＝Ａ４）に書き込みを行ったデータを、ＬＤ４がアドレスＡ４（＝Ａ４’）から読み出すことが期待されているため、それらの実行も、前にＳＴ１、その後にＬＤ４のプログラム順序で行われる必要がある。

【００２３】

図９（ｂ）は、図９（ａ）で示したプログラム順序の命令列のプログラム順序実行の例を示す図である。

【００２４】

左からサイクル番号、そのサイクルの実行命令、実行命令の対象アドレスを示す。ここで、ＳＴ１の対象アドレスＡ４’は、５サイクル目まで判明しないと想定すると、

１サイクル目にアドレスＡ１に対するロード命令ＬＤ１、

５サイクル目にアドレスＡ４’に対するストア命令ＳＴ１、

６サイクル目にアドレスＡ２に対するロード命令ＬＤ２、

７サイクル目にアドレスＡ３に対するロード命令ＬＤ３、

８サイクル目にアドレスＡ４に対するロード命令ＬＤ４、

が実行されることになる。

【００２５】

仮に２サイクル目、または３サイクル目、または４サイクル目に、アドレスＡ２、またはアドレスＡ３、またはアドレスＡ４が判明していたとしても、ＬＤ２、またはＬＤ３、またはＬＤ４を、２サイクル目、または３サイクル目、または４サイクル目に、ＳＴ１を追い越して実行する非プログラム順序実行を行うことはできない。

【００２６】

なぜなら、ＳＴ１の対象アドレスＡ４’が５サイクル目まで判明しないため、５サイクル目まで、ＳＴ１からＬＤ１、ＬＤ２、ＬＤ３、ＬＤ４への正依存関係の有無も判明しないからである。

【００２７】

すなわち、図９（ｂ）に示したプログラム順序実行の例の場合、２サイクル目、３サイクル目、４サイクル目には、ロード・ストア命令を実行することができず、ＬＤ１、ＬＤ２、ＬＤ３、ＬＤ４、ＳＴ１の５命令の実行に８サイクルを要してしまう。

【００２８】

このように、ロード・ストア命令に関して非プログラム順序実行を行わない方法では、実行性能が低下する、という問題があった。

【００２９】

この問題に対して、正依存関係の有無が判明するよりも以前に、正依存関係が存在しないと仮定して、投機的に非プログラム順序実行を行ってしまう方式が、従来より、知られている。

【００３０】

この明細書では、この方式による、データの正依存関係に対する投機的命令実行（speculative execution）を、「データ依存投機実行」と表記する、ことにする。

【００３１】

データ依存投機実行においては、

・実際に正依存関係が存在せず投機実行が成功する場合と、

・実際に正依存関係が存在して投機実行が失敗する場合と、

があり、正依存関係の有無が判明した時点で、そのいずれであるかを判定する必要がある。

【００３２】

実際に正依存関係が存在せず投機実行が成功した場合、そのまま後続の命令実行を継続することが可能であり、データ依存投機実行による非プログラム順序実行を行った分だけ、実行性能の向上が期待できる。

【００３３】

一方、実際に正依存関係が存在して投機実行が失敗した場合、プログラムの意味が変わってしまうため、プログラムの正しい実行結果が保証できなくなる。そのため、データ依存投機実行による非プログラム順序実行を行った命令の結果を取り消して、再びプログラム順序で再実行するなどのデータ依存投機実行の失敗に対する回復処理（recovery）が必要となる。投機実行が失敗した場合、失敗した命令の取り消しや回復処理のために、プログラム順序実行するよりも性能が低下することが多い。しかし、投機実行に失敗する場合よりも成功する場合の確率が十分に高ければ、プログラム全体としての実行性能の向上が期待できることになる。

【００３４】

なお、非プログラム順序実行については、マイク・ジョンソンによる文献１（

“スーパースカラ・プロセッサ”、日経ＢＰ出版センター、１９９４年）に詳しい。

【００３５】

また、投機実行の失敗による回復処理方法については、例えば、特開平５－２２４９２７号公報に開示されている方法がある。

【００３６】

データ依存投機実行の具体例を、図９を参照して説明する。図９（ｃ）は、図９（ａ）に示したプログラム順序の命令列に対して、データ依存投機実行を行いかつ投機実行に成功した場合の実行例を示す図である。

【００３７】

図９（ｂ）で示したプログラム順序実行の場合と同様に、ＳＴ１の対象アドレスＡ４’は５サイクル目まで判明しないとする。また、ＬＤ２の対象アドレスＡ２は２サイクル目で、ＬＤ３の対象アドレスＡ３は３サイクル目で、ＬＤ４の対象アドレスＡ４は６サイクル目で判明すると想定する。

【００３８】

まず、１サイクル目にアドレスＡ１に対するロード命令ＬＤ１を実行する。

【００３９】

次に２サイクル目において、プログラム順序ではＳＴ１を実行するのであるが、対象アドレスＡ４’がまだ不明であるために実行できない。このため、対象アドレスＡ２が判明しているＬＤ２を、ＳＴ１を追い越して、非プログラム順序で実行する。ここで、ＳＴ１の対象アドレスＡ４’はまだ不明であるため、ＳＴ１からＬＤ２への正依存関係の有無もまた不明である。すなわち、ＬＤ２はＳＴ１に対してデータ依存投機実行を行う。

【００４０】

同様にして、３サイクル目ではＳＴ１はまだ実行できないため、対象アドレスＡ３が判明しているＬＤ３のデータ依存投機実行を行う。このとき、ＳＴ１からＬＤ３への正依存関係の有無もまた不明である。

【００４１】

次に、４サイクル目では、ＳＴ１及びＬＤ４ともにその対象アドレスは不明で

あるので、いずれの命令も実行できない。

【００４２】

次に、５サイクル目においてＳＴ１の対象アドレスＡ４’が判明するので、ＳＴ１の実行を行う。同時に、ＳＴ１からデータ依存投機実行を行ったＬＤ２、ＬＤ３に対する正依存関係の有無を判定する。この場合、ＬＤ２の対象アドレスＡ２、及びＬＤ３の対象アドレスＡ３は、ＳＴ１の対象アドレスＡ４’と異なるため、ＬＤ２及びＬＤ３に対して正依存関係は存在しない。

【００４３】

すなわち、ＬＤ２及びＬＤ３のデータ依存投機実行は成功したと判断する。従って、後続の命令実行をそのまま継続することができ、次の６サイクル目にＬＤ４を実行する。ＬＤ４はＳＴ１から正依存関係があるが、プログラム順序で実行されているので問題は生じない。

【００４４】

以上、データ依存投機実行が成功した場合について説明した。図９（ｂ）に示したプログラム順序で実行した場合、実行に８サイクルを要したのに対して、データ依存投機実行による非プログラム順序実行が成功した場合、実行は６サイクルで完了するため、その分だけ実行性能が向上する。

【００４５】

一方、図９（ｄ）は、図９（ａ）に示したプログラム順序の命令列に対して、データ依存投機実行を行い、かつ、投機実行に失敗した場合の実行例を示している。図９（ｂ）に示したプログラム順序実行の場合と同様に、ＳＴ１の対象アドレスＡ４’は５サイクル目まで判明しないとする。

【００４６】

また、ＬＤ２の対象アドレスＡ２は２サイクル目で、ＬＤ３の対象アドレスＡ３は３サイクル目で、ＬＤ４の対象アドレスＡ４は４サイクル目で判明すると想定する。

【００４７】

まず、１サイクル目にアドレスＡ１に対するロード命令ＬＤ１を実行する。

【００４８】

次に、図９（c）に示したデータ依存投機実行が成功した場合と同様に、２サイクル目及び３サイクル目においては、プログラム順序で次の命令であるＳＴ１は、対象アドレスＡ４’がまだ不明であるために実行できないため、対象アドレスが判明しているＬＤ２及びＬＤ３をデータ依存投機実行による非プログラム順序実行を行う。

【００４９】

次に、４サイクル目では、未だＳＴ１の対象アドレスＡ４’は不明であるが、ＬＤ４の対象アドレスＡ４は判明しているため、ＬＤ４のデータ依存投機実行を行う。

【００５０】

次に、５サイクル目においてＳＴ１の対象アドレスＡ４’が判明するので、ＳＴ１の実行を行う。同時に、ＳＴ１からデータ依存投機実行を行ったＬＤ２、ＬＤ３、ＬＤ４に対する正依存関係の有無を判定する。この場合、ＬＤ２の対象アドレスＡ２、及び、ＬＤ３の対象アドレスＡ３は、ＳＴ１の対象アドレスＡ４’と異なるため、ＬＤ２及びＬＤ３に対しては、正依存関係は存在しない。

【００５１】

しかし、ＬＤ４の対象アドレスＡ４は、ＳＴ１の対象アドレスＡ４’に等しいため、ＳＴ１からＬＤ４に対して正依存関係が存在する。すなわち、正依存関係が存在するにもかかわらず、非プログラム順序で実行を行ってしまっているので、ＬＤ４のデータ依存投機実行は失敗したと判断する。

【００５２】

この場合、このまま後続の命令の実行を継続すると、プログラムの正しい実行結果が保証されないので、データ依存投機実行の失敗の回復処理の実行が必要である。

【００５３】

例えば、図９（ｄ）に示した実行例では、５サイクル目においてデータ依存投機実行は失敗したと判断されると、まず、データ依存投機実行を行った２サイクル目から４サイクル目のＬＤ２、ＬＤ３、ＬＤ４及び５サイクル目のＳＴ１の実行結果を取り消し、次に、実行結果を取り消したＳＴ１、ＬＤ２、ＬＤ３、ＬＤ

４の再実行を７サイクル目からプログラム順序で実行することで、データ依存投機実行の失敗の回復を行い、その後、後続命令の実行を再開する。

【００５４】

この場合、ＬＤ１、ＬＤ２、ＬＤ３、ＬＤ４、ＳＴ１の５命令の実行に、１０サイクルを要することになり、プログラム順序で実行を行った場合の８サイクルより、性能が低下することになる。

【００５５】

しかしながら、データ依存投機実行に成功した場合には、６サイクルで実行できるため、投機実行に失敗する場合よりも成功する場合の確率が十分に高ければ、プログラム全体としての実行性能の向上が期待できる。

【００５６】

以上述べたように、プロセッサがデータ依存投機実行を行うには、ロード・ストア命令間のメモリに対する正依存関係の有無を判定する機能が必要となる。この機能を担う装置として、例えば、Manoj Franklinらによる論文"ARB: A Hardware Mechanism for Dynamic Reordering of Memory References", IEEE Transactions on Computers, vol.45, no.5(1996年5月)に開示されているデータ依存関係検出装置がある。

【００５７】

この従来のデータ依存関係検出装置の構成及び動作を図１０を参照して説明する。図１０は、従来のデータ依存関係検出装置１００の構成を示すブロック図である。図１０において、符号１０１はアドレスバッファ、符号１０２はアドレス比較器、符号１０３は論理和回路である。データ依存関係検出装置１００は、ロード命令の対象アドレスを格納する複数のアドレスバッファ１０１と、各々のアドレスバッファに接続された複数のアドレス比較器１０２と、すべてのアドレス比較器１０２の出力を入力とする論理和回路１０３とから構成される。

【００５８】

アドレス比較器１０２はアドレスバッファ１０１に格納されたロード命令の対象アドレスと、実行中のストア命令の対象アドレスとの比較を行う。

【００５９】

論理和回路１０３は、すべてのアドレス比較器１０２の出力の論理和を演算し、データ依存検出結果として出力する。

【００６０】

データ依存関係検出装置１００によるストア命令からロード命令へのメモリに関する正依存関係の検出は、以下のような動作で実現される。まず、ロード命令がデータ依存投機実行された場合、その対象アドレスを、これまでロード命令の対象アドレスが格納されていない空いているアドレスバッファ１０１に格納していく。

【００６１】

次に、ストア命令が実行されると、その対象アドレスは、すべてのアドレス比較器１０２に入力され、アドレスバッファ１０１に格納されている、それ以前にデータ依存投機実行されたロード命令の対象アドレスと比較する。

【００６２】

アドレス比較器１０２から出力された比較結果はすべて論理和をとり、データ依存検出結果して出力する。ストア命令の対象アドレスと、アドレスバッファ１０１に格納されているロード命令の対象アドレスのいずれとも一致しない場合、そのストア命令からアドレスバッファ１０１に対象アドレスを格納したロード命令へ、正依存関係が存在しないと判断することができるので、その旨をデータ依存検出結果して出力する。この場合、データ依存投機実行が成功したとして、そのまま後続の命令の実行を継続することができる。

【００６３】

一方、ストア命令の対象アドレスと、アドレスバッファ１０１に格納されているロード命令の対象アドレスのいずれかが一致した場合、そのストア命令からアドレスバッファ１０１に対象アドレスを格納したロード命令へ、正依存関係が存在すると判断することができるので、その旨をデータ依存検出結果して出力する。この場合、データ依存投機実行が失敗したとして、データ依存投機実行が失敗したことによる回復処理などが行われることになる。

【００６４】

上述した従来のデータ依存関係検出装置１００には、次のような問題点がある

。

【００６５】

第１の問題点は、ハードウエア量が大きいことである。データ依存投機実行においてプログラムの実行結果が正しく得られることを保証するには、すべての正依存関係をもれなく検出することが必要である。

【００６６】

従来のデータ依存関係検出装置１００では、データ依存投機実行を行ったすべてのロード命令の対象アドレスをアドレスバッファ１０１に格納し、アドレス比較器１０２により、後続のストア命令の対象アドレスと比較する必要がある。そのため、すべてのアドレスバッファ１０１に、それまでデータ依存投機実行したロード命令の対象アドレスが既に格納されて、空いているアドレスバッファ１０１がない場合には、それ以上ロード命令をデータ依存投機実行することができない。

【００６７】

この場合、それ以降のロード・ストア命令はプログラム順序で実行されることになるため、データ依存投機実行することが可能なロード命令数は、アドレスバッファ１０１の設置数及びそれと同数のアドレス比較器１０２の設置数に制限されることになる。従って、データ依存投機実行による実行性能の向上を享受するためには、充分な数のアドレスバッファ１０１及びアドレス比較器１０２を設置する必要があるため、必要なハードウエア量が大きくなってしまうという問題がある。

【００６８】

第２の問題点は、正依存関係の検出処理の処理速度が遅いことである。従来のデータ依存関係検出装置１００では、正依存関係の検出処理に、アドレス比較器１０２によるアドレス比較の処理時間に加えて、論理和回路１０３によるすべてのアドレス比較器１０２からの出力の論理和の演算時間が必要なため、動作速度が遅く、その結果、プロセッサの動作周波数を高速化することが困難であるいう問題点がある。

【００６９】

この問題は、アドレスバッファ１０１及びアドレス比較器１０２の設置数が多くなるにつれて、論理和回路１０３の入力数が増加するため、益々顕著になる。すなわち、第１の問題点で述べたように、従来のデータ依存関係検出装置１００では、データ依存投機実行による実行性能の向上を享受するために、必要なハードウエア量が大きくなってしまうが、その場合、正依存関係検出処理の処理時間が益々増加し、プロセッサの動作周波数を高速化することが困難になるため、結局は実行性能が低下してしまうという問題がある。

【００７０】

【発明が解決しようとする課題】

本発明は、このような事情に鑑みてなされたもので、その目的は、データ依存投機実行することのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されず、小さいハードウエア量でもデータ依存投機実行による実行性能の向上を享受することのできる、ハードウエア量が小さいデータ依存関係検出装置を提供することにある。

【００７１】

また、本発明の別の目的は、正依存関係の検出の処理時間が短く、プロセッサの動作周波数を高速化することが容易である、高速なデータ依存関係検出装置を提供することにある。

【００７２】

【課題を解決するための手段】

上記目的を達成するために、本発明は、プロセッサが非プログラム順序でロード命令またはストア命令の実行を行う際に、前記ロード命令またはストア命令が処理対象とするアドレス間の依存関係の存在の可能性を検出するデータ依存関係検出装置であって、実際に前記依存関係が存在する場合は必ず依存関係が存在する旨を検出するが、実際に前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容する構成としたものであり、ロード命令及びストア命令が処理対象とするアドレスを一意な番号に変換するアドレス変換手段と、前記変換された番号に対応して、ロード命令又はストア命令が実行されたか否かを記憶する複数の記憶手段から構成される実行履歴記憶手段とを少なくと

も含むことを特徴とする。

【００７３】

上記構成によれば、実行されたロード命令またはストア命令が処理対象とするアドレス自体をすべて記憶しておく必要がなく、また前記アドレスを比較するためのアドレス比較器も必要としない。

【００７４】

すなわち、本発明のデータ依存関係検出装置によれば、データ依存投機実行の成否判定に必要なハードウエア量が小さいという効果を得ることができる。さらに、データ依存投機実行することのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもデータ依存投機実行による実行性能の向上を享受することのできるという効果を得ることができる。また、ロード命令、ストア命令の対象アドレス間の依存関係の検出処理の際に、ロード命令またはストア命令の対象アドレスの比較を必要しない。さらに、前記実行履歴手段を構成する複数の記憶手段のうち、読み出しの対象となる記憶手段は高々１つだけで充分なので、依存関係の検出の処理時間が短く、プロセッサの動作周波数を高速化することが容易であるという効果を得ることができる。

【００７５】

また、本発明は、ロード命令及びストア命令などのメモリ操作命令を実行可能な複数のプロセッサから構成されるマルチプロセッサシステムにおいて、前記各々のプロセッサが備えるデータ依存関係検出装置であって、かつ前記マルチプロセッサシステムがスレッド単位で並列処理を行う際に、他のプロセッサが実行するスレッドが含むロード命令またはストア命令が処理対象とするアドレスと、自プロセッサが実行するスレッドが含むロード命令またはストア命令が処理対象とするアドレスとの間に依存関係が存在する可能性を検出する機能を有するデータ依存関係検出装置であって、前記データ依存関係検出装置は、実際に前記依存関係が存在する場合は必ず依存関係が存在する旨を検出するが、実際に前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容することを特徴とし、自プロセッサ及び他プロセッサが実行するロード命令及び

ストア命令が処理対象とするアドレスを一意な番号に変換する複数のアドレス変換手段と、前記変換された番号に対応して、ロード命令又はストア命令が実行されたか否かを記憶する複数の記憶手段から構成される実行履歴記憶手段とを少なくとも含むことを特徴とする。

【００７６】

上記構成によれば、スレッド並列処理におけるスレッド単位のデータ依存投機実行においても、データ依存投機実行の成否判定に必要なハードウエア量が小さいという効果を得ることができる。さらに、データ依存投機実行を行うことのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもスレッド単位のデータ依存投機実行による実行性能の向上を享受することのできるという効果を得ることができる。さらに、スレッド間の依存関係の検出の処理時間が短く、プロセッサの動作周波数を高速化することが容易であるという効果を得ることができる。

【００７７】

また、本発明によるデータ依存関係検出装置は、ロード命令及びストア命令などのメモリ操作命令を実行可能な複数のプロセッサから構成されるマルチプロセッサシステムにおいて、前記各々のプロセッサが備えるデータ依存関係検出装置であって、前記マルチプロセッサシステムがスレッド単位で並列処理を行う際に、他のプロセッサが実行するスレッドが含むロード命令またはストア命令が処理対象とするアドレスと、自プロセッサが実行するスレッドが含むロード命令またはストア命令が処理対象とするアドレスとの間に依存関係が存在する可能性を検出する機能、及び自プロセッサが実行するスレッドが含むロード命令またはストア命令が対象とするアドレス間の依存関係が存在する可能性を検出する機能を有するデータ依存関係検出装置であって、前記データ依存関係検出装置は、実際に前記依存関係が存在する場合は必ず依存関係が存在する旨を検出するが、実際に前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容することを特徴とし、自プロセッサ及び他プロセッサが実行するロード命令及びストア命令が処理対象とするアドレスを一意な番号に変換する複数のアドレス変換手段と、前記変換された番号に対応して、ロード命令又はストア命

令が実行されたか否かを記憶する複数の記憶手段から構成される実行履歴記憶手段とを少なくとも含むことを特徴とする。

【００７８】

上記構成によれば、スレッド並列処理におけるスレッド単位のデータ依存投機実行及び、スレッド内の命令単位のデータ依存投機実行を可能とし、さらにそれらデータ依存投機実行の成否判定に必要なハードウエア量が小さいという効果を得ることができる。さらに、データ依存投機実行することのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもスレッド単位のデータ依存投機実行及びスレッド内の命令単位のデータ依存投機実行による実行性能の向上を享受することのできるという効果を得ることができる。さらに、データ依存関係の検出の処理時間が短く、プロセッサの動作周波数を高速化することが容易であるという効果を得ることができる。

【００７９】

【発明の実施の形態】

次に、本発明の実施の形態について図面を参照して詳細に説明する。

【００８０】

図１、は本発明の第１の実施の形態のデータ依存関係検出装置を含むプロセッサの構成を示すブロック図である。図１において、符号１０は本発明の第１の実施の形態によるデータ依存関係検出装置、符号１７はプロセッサ制御部、符号１８は命令実行部である。プロセッサの他の構成要素については図示していない。

【００８１】

プロセッサが命令を実行する場合、プロセッサ制御部１７は、実行する命令の種別を命令実行部１８、及び、データ依存関係検出装置１０に出力する。

【００８２】

データ依存関係検出装置１０は、その実行する命令種別を実行命令１４から入力する。

【００８３】

さらに、実行する命令がロード命令またはストア命令の場合、プロセッサ制御

部１７は、その対象アドレスを命令実行部１８及びデータ依存関係検出装置１０に出力する。

【００８４】

データ依存関係検出装置１０は、そのロード・ストア命令の対象アドレスをロード・ストアアドレス１５から入力する。

【００８５】

さらに、ロード命令またはストア命令をデータ依存投機実行する場合、プロセッサ制御部１７は、その旨を投機実行フラグ１３を通じてデータ依存関係検出装置１０に通知する。

【００８６】

データ依存関係検出装置１０は、ハッシュ関数回路１１、及び命令履歴テーブル１２を備えており、プロセッサ制御部１７からの入力として、投機実行フラグ１３、実行命令１４、ロード・ストアアドレス１５を有し、プロセッサ制御部１７への出力としてデータ依存検出結果１６を有する。

【００８７】

ハッシュ関数回路１１は、ｍビットで表現されるロード命令またはストア命令の対象アドレスを、ｎビットで表現される命令履歴テーブル１２のエントリ番号に変換するハッシュ関数ｆを実現する論理回路である。通常、ｍよりもｎが小さい（ｍ＞ｎ）。ここで、ハッシュ関数ｆは、同一の入力に対しては、同一の値をもつ。すなわち、Ｎ１＝ｆ（Ａ１）かつＮ２＝ｆ（Ａ２）としたとき、Ａ１＝Ａ２であれば、Ｎ１＝Ｎ２が成立する関数である。従って、ハッシュ関数回路１１は、同一のロード・ストア命令の対象アドレスに対しては、必ず同一の命令履歴テーブルのエントリ番号を出力することを保証する。

【００８８】

一方、Ａ１≠Ａ２であっても、一般にはＮ１≠Ｎ２は保証されない。すなわち、異なるロード・ストア命令の対象アドレスに対して、同一の命令履歴テーブルのエントリ番号が出力される場合が存在する。

【００８９】

命令履歴テーブル１２は、１ビットの情報を格納できる複数のエントリから構

成される。各エントリはメモリやフリップフロップなどの記憶手段で実現することができる。命令履歴テーブル１２が備える上記エントリの総数は、ハッシュ関数回路１１の出力がｎビット幅である場合、２のｎ乗に等しい数である。命令履歴テーブル１２は、ハッシュ関数回路１１が出力するエントリ番号が指し示すエントリに対して書き込み及び読み出し処理を行う。

【００９０】

さらに、プロセッサ制御部１７からデータ依存関係検出装置１０に入力される投機実行フラグ１３に基づいて、すべてのエントリに同一の値を書き込む初期化機能を有する。

【００９１】

上記構成のデータ依存関係検出装置１０の動作について、図１及び図２を参照して説明する。

【００９２】

図２は、データ依存関係検出装置１０の動作を示すタイミング図である。まず、初期状態においては、命令履歴テーブル１２を構成するすべてのエントリは、ロード命令が実行されていないことを示す第１の状態（例えば論理０）にある。プロセッサがデータ依存関係に対して投機的な命令実行を行わず、プログラム順序で確定的に命令実行を行う状態にある場合（この状態を「確定実行状態」という）、データ依存関係検出装置１０はなんら動作を行わない。図２においては、サイクル０からサイクル３までが確定実行状態である。

【００９３】

それに対して、プロセッサがデータ依存関係に対して投機的な命令実行を行う状態（この状態を「投機実行状態」という）にある場合、データ依存関係検出装置１０は、ストア命令からロード命令への正依存関係の検出処理を行う。図２においては、サイクル４から８までが投機実行状態である。

【００９４】

プロセッサが確定実行状態または投機実行状態のどちらにあるかは、プロセッサ制御部１７からデータ依存関係検出装置１０に入力される投機実行フラグ１３で判断される。図２においては、投機実行フラグ１３の値が論理０である場合、

プロセッサは確定実行状態にあることを示し、一方、投機実行フラグ１３の値が論理１である場合、プロセッサは投機実行状態にあることを示す。

【００９５】

投機実行状態において、投機的なロード命令が実行されると（図２のサイクル５）、データ依存関係検出装置１０は、投機的なロード命令が実行されたことを命令履歴テーブル１２に記憶する。より詳細には、ロード命令の対象アドレスをＡ１とすると、その対象アドレスＡ１をハッシュ関数回路１１によりエントリ番号Ｎ１に変換し、命令履歴テーブル１２に入力する。命令履歴テーブル１２は、入力されたエントリ番号Ｎ１に対応するエントリを、投機的なロード命令が実行されたことを示す第２の状態（例えば論理１）に変更する。

【００９６】

一方、投機的なストア命令が実行された場合（図２のサイクル７）には、データ依存関係検出装置１０の対応するエントリに記憶された状態の読み出しを行う。より詳細には、ストア命令の対象アドレスをＡ２とすると、その対象アドレスＡ２をハッシュ関数回路１１によりエントリ番号Ｎ２に変換し、命令履歴テーブル１２に入力する。命令履歴テーブル１２は、入力されたエントリ番号Ｎ２に対応するエントリの内容を読み出し、その内容に基づいてデータ依存検出結果１６からプロセッサ制御部１７に出力する。

【００９７】

例えば図２において、サイクル５で実行されたロード命令の対象アドレスＡ１と、サイクル７で実行されたストア命令の対象アドレスＡ２が等しい場合（Ａ１＝Ａ２）、ハッシュ関数回路１１の性質により、それぞれに対応するエントリ番号Ｎ１、Ｎ２は互いに等しい（Ｎ１＝Ｎ２）。すなわち、サイクル５にロード命令によって命令履歴テーブル１２のエントリＮ１に格納された値（論理１）は、サイクル７で実行されたストア命令によって命令履歴テーブル１２から読み出されるため、サイクル７で実行されたストア命令からサイクル５で実行されたロード命令へ、正依存関係が存在する可能性があることが検出される。

【００９８】

ここで検出されるのが、“正依存関係が存在すること”ではなく、“正依存関

係が存在する可能性があること”であるのは、後述するエイリアスの発生する場合があるために、ストア命令によって命令履歴テーブル１２から投機的なロード命令が実行されたことを示す値が読み出された場合でも、実際には正依存関係が存在しない場合があるためである。

【００９９】

いずれにせよ、正依存関係が存在する可能性があることが検出された場合、データ依存関係検出装置１０は、データ依存投機実行が失敗したことを示す値（例えば論理１）をデータ依存検出結果１６に出力する。その場合、プロセッサ制御部１７及び命令実行部１８はデータ依存投機実行が失敗したことに対する回復処理を実行する。

【０１００】

一方、サイクル５で実行されたロード命令の対象アドレスＡ１と、サイクル７で実行されたストア命令の対象アドレスＡ２が異なる場合（Ａ１≠Ａ２）、ハッシュ関数回路１１の性質により、それぞれに対応するエントリ番号Ｎ１、Ｎ２は互いに異なる場合（Ｎ１≠Ｎ２）と、互いに等しい場合（Ｎ１＝Ｎ２）の両方が存在する。前者の、エントリ番号Ｎ１、Ｎ２が互いに異なる場合、サイクル７のストア命令実行時には、命令履歴テーブル１２のエントリＮ２の値は初期値である論理０であるため、サイクル７で実行されたストア命令からサイクル５で実行されたロード命令へ、正依存関係が存在しないことが検出される。この場合、データ依存投機実行が成功したことを示す値（例えば論理０）をデータ依存検出結果１６に出力する。プロセッサ制御部１７はデータ依存投機実行が成功したとして、データ依存投機実行が失敗したことに対する回復処理は実行せず、後続の命令実行をそのまま続行することができる。

【０１０１】

後者の、エントリ番号Ｎ１、Ｎ２が互いに等しい場合、サイクル５のロード命令とサイクル７のストア命令は、同じ命令履歴テーブル１２のエントリＮ１（＝Ｎ２）を参照する。すなわち、ロード命令・ストア命令それぞれの対象アドレスＡ１、Ａ２が互いに異なり、ストア命令からロード命令へ正依存関係が存在しないのにもかかわらず、サイクル７のストア命令実行時には、サイクル５でロード

命令により書き込まれた論理１が命令履歴テーブル１２から読み出されるため、サイクル７で実行されたストア命令からサイクル５で実行されたロード命令へ、正依存関係が存在する可能性があることが検出される。この場合、データ依存投機実行が失敗したことを示す値（例えば論理１）をデータ依存検出結果１６に出力する。プロセッサ制御部１７及び命令実行部１８は、データ依存投機実行が失敗したことに対する回復処理を実行する。上記したように、互いに異なる対象アドレスが、命令履歴テーブル１２の同じエントリに割り当てられることを、以下では“エイリアスが発生する”と表記する。

【０１０２】

ロード命令とストア命令の間にエイリアスが発生した場合、正依存関係が存在しない場合であっても、プロセッサ制御部１７には正依存関係が存在した旨が通知され、本来不必要なデータ依存投機実行失敗による回復処理が実行されてしまう。しかし、この場合でも回復処理は不必要ではあるが、プログラム上の意味を変更してしまうことはないため、プログラムの実行結果の正しさは保証される。

【０１０３】

一方で、不必要な回復処理が実行されることにより、プログラムの実行性能が低下するという問題がある。しかし、この問題に対しては後述するように、ハッシュ関数回路１１のハッシュ関数を適切に選択したり、命令履歴テーブル１２のエントリを十分多く実装することにより、エイリアスの発生確率は低減することができる。エイリアスの発生確率が充分小さければ、データ依存投機実行の効果により、プログラム全体の実行性能の向上を期待することができる。

【０１０４】

エイリアスはロード命令の間にも発生する。例えば、対象アドレスがＡ１であるロード命令ＬＤ１、及び対象アドレスがＡ２であるロード命令ＬＤ２の２つのロード命令が実行されたとき、それぞれの対象アドレスが異なっている（Ａ１≠Ａ２）場合でも、参照されるそれぞれの命令履歴テーブル１２のエントリN1及びＮ２が互いに同じになる（Ｎ１＝Ｎ２）エイリアスが発生する場合がある。この場合、命令履歴テーブル１２の同じエントリN1（＝Ｎ２）にロード命令が実行されたことを示す論理１が格納される。

【０１０５】

次に、ＬＤ１またはＬＤ２に正依存関係のある対象アドレスがＡ１またはＡ２のストア命令が実行された場合、命令履歴テーブル１２のエントリN1（＝Ｎ２）が参照され、正依存関係の存在が検出されるが、ＬＤ１、ＬＤ２のどちらに対して正依存関係が存在するかは区別することができない。

【０１０６】

しかし、正依存関係が存在すれば必ず検出され、データ依存投機実行失敗による回復処理が実行されるため、この場合も、プログラムの実行結果の正しさは保証される。

【０１０７】

上述したように、本発明によるデータ依存関係検出装置は、ストア命令からロード命令への正依存関係の存在を厳密に検出するのではなく、ストア命令からロード命令への正依存関係の存在の可能性を検出する、ことを主たる特徴の一つとしている。

【０１０８】

任意のロード命令の対象アドレスは、ハッシュ関数回路１１により、命令履歴テーブル１２の特定のエントリに重複を許して割り当てるため、従来のデータ依存関係検出装置と異なり、実行されたロード命令の対象アドレス自体をすべて記憶しておく必要がなく、またストア命令の対象アドレスと比較するためのアドレス比較器も必要としない。

【０１０９】

すなわち、この実施の形態によるデータ依存関係検出装置によれば、データ依存投機実行の成否判定に必要なハードウエア量が小さいという効果を得ることができる。さらに、この実施の形態によるデータ依存関係検出装置によれば、データ依存投機実行することのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもデータ依存投機実行による実行性能の向上を享受することのできるという効果を得ることができる。

【０１１０】

また、この実施の形態によるデータ依存関係検出装置によれば、ストア命令の実行時の正依存関係の検出処理において、ストア命令の対象アドレスとロード命令の対象アドレスの比較を必要しない。

【０１１１】

さらに、命令履歴テーブルの読み出すエントリは１つだけなので、すべてのエントリ出力の論理和を演算する必要もない。すなわち、この実施の形態によるデータ依存関係検出装置によれば、正依存関係の検出の処理時間が短いため、プロセッサの動作周波数を高速化することが容易であるという効果を得ることができる。

【０１１２】

次に、本発明の第１の実施の形態によるデータ依存関係検出装置の動作の具体例を説明する。

【０１１３】

図３は、ハッシュ関数回路２１及び命令履歴テーブル２２から構成されるデータ依存関係検出装置２０の構成を模式的に示すブロック図である。図３のハッシュ関数回路２１、命令履歴テーブル２２、データ依存関係検出装置２０は、図１のハッシュ関数回路１１、命令履歴テーブル１２、データ依存関係検出装置１０にそれぞれ対応している。実行命令２４、投機実行フラグ２３、データ依存検出結果２６は、図１の実行命令１４、投機実行フラグ１３、データ依存検出結果１６にそれぞれ対応している。プロセッサ制御部（図３では図示せず、図１の１７）からの入力として、投機実行フラグ、実行命令、ロード・ストアアドレスを備える。また、プロセッサ制御部への出力として、データ依存検出結果を備える。ここで、ロード・ストア命令の対象アドレスは８ビットで表現されるものとする。

【０１１４】

ハッシュ関数回路２１の第１の実施例を、図５（ａ）に示す。図５（ａ）を参照すると、８ビットで与えられるロード・ストア命令の対象アドレスの２ビット目及び３ビット目をそのまま取り出すことで、２ビットで表現される命令履歴テーブル２２のエントリ番号に変換する。命令履歴テーブル２２は、４つのエント

リから構成され、ハッシュ関数回路２１が出力する２ビット表現のエントリ番号、及び、プロセッサ制御部が出力する実行命令と投機実行フラグを参照して、各エントリの書き込みや読み出し処理を行う。

【０１１５】

図４を参照して、データ依存関係検出装置２０の動作の具体例を以下に説明する。まず、図４（ａ）はプログラム例における命令のプログラム順序を示しており、

ロード命令ＬＤ１、

ストア命令ＳＴ１、

ロード命令ＬＤ２、

ロード命令ＬＤ３、

ロード命令ＬＤ４

の順序である。ここで、ＬＤ１の対象アドレスはＡ１、ＬＤ２の対象アドレスはＡ２、ＬＤ３の対象アドレスはＡ３、ＬＤ４の対象アドレスはＡ４、ＳＴ１の対象アドレスはＡ４である。

【０１１６】

ＬＤ４とＳＴ１の対象アドレスは同じＡ４であり、プログラム順序でＳＴ１の方が前にあるため、ＳＴ１からＬＤ４にはデータ依存関係が存在する。すなわち、実行時においてＬＤ４よりもＳＴ１が前に実行されて、ＳＴ１がアドレスＡ４に書き込みを行った結果を、ＬＤ１がアドレスＡ４から読み出しを行わなければ、プログラムの実行結果の正しさは保証されない。

【０１１７】

プロセッサがデータ依存投機実行状態にある場合、ストア命令からロード命令への正依存関係の有無が不明であっても、実行可能になった命令から実行される。

【０１１８】

図４（ｂ）は、データ依存投機実行状態における、図４（ａ）に示したプログラム順序の命令列のデータ依存投機実行による非プログラム順序実行例であり、左から順にサイクル番号、そのサイクルの実行命令、その実行命令の対象アドレ

スをそれぞれ示す。すなわち、０サイクル目にＬＤ１、１サイクル目にＬＤ２、２サイクル目にＬＤ２、３サイクル目にＬＤ３、４サイクル目にＬＤ４、５サイクル目にＳＴ１、が実行されている。各実行命令の対象アドレスは、２進数８ビットで表記しており、Ａ１＝００１００００、Ａ２＝００１０１０００、Ａ３＝００００１１００、Ａ４＝００１００００、である。

【０１１９】

正依存関係にあるＳＴ１及びＬＤ４について、５サイクル目のＳＴ１よりも後に実行されるべきＬＤ４が、それより前の４サイクル目に実行されているため、正依存関係に違反しており、データ依存投機実行に失敗している。すなわち、ＬＤ４はアドレスＡ４から、ＳＴ１がアドレスＡ４に書き込みを行った値ではなく、ＳＴ１が書き込みを行う以前の値を読み出してしまうため、プログラムの正しい実行結果を得ることが保証されないことになる。

【０１２０】

データ依存関係検出装置２０（図３参照）は、このデータ依存投機実行の失敗を検出し、プロセッサ制御部１７(図１参照)へデータ依存検出結果１６（図１参照）の出力によって通知する機能を担う。この通知により、プロセッサ制御部は、正しいプログラムの実行結果が選られるように、データ依存投機実行した命令の再実行などのデータ依存投機実行の失敗による回復処理を行う。

【０１２１】

図４（ｃ）は、図４（ｂ）に示した命令実行順序に対するデータ依存関係検出装置２０の動作を示す図である。左から順にサイクル番号、そのサイクルの投機実行フラグ（図１の１３）の値、そのサイクルに実行されたロード・ストア命令が参照する命令履歴テーブル２２のエントリ番号、そのサイクルの命令履歴テーブル２２の状態をそれぞれ示す。命令履歴テーブル２２の状態としては、図の右から順に、エントリ００、エントリ０１、エントリ１０、及び、エントリ１１の４つのエントリの内容が示されている。

【０１２２】

例えば、０サイクル目は投機実行フラグの値は論理０であり、またロード・ストア命令は実行されていない。命令履歴テーブル２２は初期状態にあり、４つの

エントリの内容はすべて、ロード命令が実行されていないことを示す論理０である。

【０１２３】

次に１サイクル目から５サイクル目においてプロセッサは投機実行状態になり、その旨を示す投機実行フラグが論理１になる。

【０１２４】

１サイクル目では、ロード命令ＬＤ１がデータ依存投機実行される。ハッシュ関数回路２１は、ＬＤ１の対象アドレス　Ａ１＝００１００００００の２ビット目及び３ビット目から、命令履歴テーブル２２に参照するエントリの番号００を出力する。命令履歴テーブル２２は、ハッシュ関数回路２１が出力するエントリ番号００、及び、プロセッサ制御部が出力する実行命令と投機実行フラグを参照して、エントリ００の内容を、ロード命令が実行されたことを示す論理１にする。

【０１２５】

次に２サイクル目において、ロード命令ＬＤ２がデータ依存投機実行される。ハッシュ関数回路２１は、ＬＤ２の対象アドレス　Ａ１＝００１０１０００の２ビット目及び３ビット目から、命令履歴テーブル２２に対して、参照エントリの番号１０を出力する。命令履歴テーブル２２は、ハッシュ関数回路２１が出力するエントリ番号１０、及び、プロセッサ制御部が出力する実行命令と投機実行フラグを参照して、エントリ１０の内容をロード命令が実行されたことを示す論理１を書き込む。

【０１２６】

同様にして、３サイクル目においては、ロード命令ＬＤ３がデータ依存投機実行され、その対象アドレスがＡ３＝０００００１１００であることから、命令履歴テーブル２２のエントリ１１に論理１を書き込む。

【０１２７】

４サイクル目においては、ロード命令ＬＤ４がデータ依存投機実行され、その対象アドレスがＡ４＝０００１００００であることから、命令履歴テーブル２２のエントリ００に論理１を書き込む。

【０１２８】

５サイクル目においては、ストア命令ＳＴ１がデータ依存投機実行され、その対象アドレスがＡ４＝０００１００００であることから、命令履歴テーブル２２のエントリ００の内容が読み出される。

【０１２９】

命令履歴テーブル２２のエントリ００には、１サイクル目と４サイクル目に論理１が書き込まれているため、エントリ００からは論理１が読み出される。すなわち、ストア命令ＳＴ１から、ロード命令ＬＤ１またはＬＤ４への正依存関係が検出される。

【０１３０】

データ依存投機実行状態にあるため、データ依存関係検出装置２０は、データ依存検出結果２６を論理１とし、プロセッサ制御部（図１の１７）に対して、データ依存投機実行が失敗したことを通知する。

【０１３１】

データ依存投機実行の失敗により、次の６サイクル目からプロセッサ制御部（図１の１７）は、データ依存投機状態中の実行命令の結果の破棄や命令の再実行等、データ依存投機実行の失敗による回復処理を行う。また、データ依存投機実行が失敗により終了して、投機実行フラグ２３が再び論理０に戻るので、命令履歴テーブル２２は、すべてのエントリの内容を初期状態である論理０でクリアして、次のデータ依存投機実行に備える。

【０１３２】

以上、本発明の第１の実施の形態によるデータ依存関係検出装置における、データ依存投機実行が失敗した場合の動作の具体例を説明した。一方、正依存関係が検出されずデータ依存投機実行が成功した場合には、データ依存投機実行の失敗による回復処理は行わず、そのまま後続の命令の実行を継続することができる。

【０１３３】

図４（ｃ）に示したデータ依存関係検出装置２０の動作において、ＬＤ１の対象アドレスＡ１＝００１００００００と、ＬＤ４の対象アドレスＡ４＝０００１０００００は互いに異なるのにもかかわらず、命令履歴テーブル２２の同じエントリ

００が参照されるエイリアスが発生している。

【０１３４】

これは、ハッシュ関数回路２１がアドレスＡ１及びアドレスＡ４を同じエントリ番号００に変換していることが原因である。従って、仮に、ストア命令ＳＴ１が実行されるより以前にロード命令ＬＤ４が実行されず、ストア命令ＳＴ１からロード命令ＬＤ４に対して正依存関係が検出されなかった場合であっても、ストア命令ＳＴ１からロード命令ＬＤ１に対して、偽の正依存関係が検出されてしまうため、データ依存投機実行は成功しているにもかかわらず、失敗と判断されてしまう。そのため、本来実行する必要のないデータ依存投機実行失敗による回復処理が実行されてしまう。データ依存投機実行失敗による回復処理は、プログラム上の意味を変更してしまうことはないため、エイリアスが発生した場合でも、プログラムの実行結果の正しさは保証される。

【０１３５】

しかし、このように、エイリアスが発生すると、不必要な回復処理が実行されるので、プログラムの実行性能が低下するという問題がある。このエイリアスによる性能低下の問題に対しては、ハッシュ関数回路２１のハッシュ関数を適切に選択したり、命令履歴テーブル２２のエントリを十分多く実装することにより、エイリアスの発生確率を十分小さくすることで低減が可能である。

【０１３６】

例えば、命令履歴テーブル２２のエントリ数を、４から８に増加し、ハッシュ関数回路２１にアドレスの２ビット目、３ビット目、４ビット目を取り出す第２の実施例（図５（ｂ））を使用した場合、アドレスＡ１＝００１００００はエントリ０００へ、アドレスＡ２＝００１０１０００はエントリ０１０へ、アドレスＡ３＝００００１１００はエントリ０１１へ、アドレスＡ４＝０００１００００はエントリ１００へ割り当てられる。すなわち、エイリアスが発生しない。

【０１３７】

あるいは、命令履歴テーブル２２のエントリ数は４のままである場合でも、ハッシュ関数回路２１に、出力するエントリ番号の０ビット目を入力されるアドレスの２ビット目と４ビット目の排他的論理和、出力するエントリ番号の１ビット

目を入力されるアドレスの３ビット目と５ビット目の排他的論理和とする第３の実施例（図５（ｃ）参照）を使用すると、アドレスＡ１＝００１００００００はエントリ１０へ、アドレスＡ２＝００１０１０００はエントリ００へ、アドレスＡ３＝０００００１１００はエントリ１１へ、アドレスＡ４＝０００１００００はエントリ０１へ割り当てられる。すなわち、エイリアスが発生しない。

【０１３８】

このように、エイリアスの発生確率は低減させることが可能であり、その場合、データ依存投機実行の効果により、プログラム全体の実行性能の向上を期待することができる。

【０１３９】

以上説明したように、本発明によるデータ依存関係検出装置によれば、データ依存投機実行の成否判定に必要なハードウエア量が小さいという効果がある。これは、従来のデータ依存関係検出装置と異なり、実行されたロード命令の対象アドレス自体をすべて記憶しておく必要がなく、またストア命令の対象アドレスと比較するためのアドレス比較器も必要としないことによる。

【０１４０】

さらには、エイリアスの存在を容認することにより、データ依存投機実行することのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもデータ依存投機実行による実行性能の向上を享受することができるためである。

【０１４１】

本発明の第１の実施の形態によるデータ依存関係検出装置による別の作用効果としては、データ依存投機実行の成否判定が高速に行えるため、プロセッサの動作周波数を高速化することが容易であることである。これは、ストア命令の実行時の正依存関係の検出処理において、ストア命令の対象アドレスとロード命令の対象アドレスの比較を必要とせず、また、命令履歴テーブルの読み出すエントリは１つだけなので、すべてのエントリ出力の論理和を演算する必要もないことによる。

【０１４２】

上記した本発明の第１の実施の形態は、ロード命令が実行された場合には、該ロード命令が処理対象とするアドレスをハッシュ関数回路１１により命令履歴テーブル１２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリに、ロード命令が実行されたことを示す値を格納し、ストア命令が実行された場合には、該ストア命令が処理対象とするアドレスをハッシュ関数回路１１により命令履歴テーブル１２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリからロード命令が実行されたか否かを示す値を読み出すことで、ストア命令からロード命令への正依存関係が存在する可能性を検出することを特徴としている。しかしながら、この実施の形態は、正依存関係が存在する可能性の検出にとどまらず、逆依存関係や出力依存関係が存在する可能性の検出に対しても、同様の構成で容易に適用することができる。

【０１４３】

例えば、ストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路１１により命令履歴テーブル１２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリに、ストア命令が実行されたことを示す値を格納し、ロード命令が実行された場合には、該ロード命令の対象アドレスをハッシュ関数回路１１により命令履歴テーブル１２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリからストア命令が実行されたか否かを示す値を読み出すことで、ロード命令からストア命令への逆依存関係が存在する可能性を検出する機能を有するデータ依存関係検出装置が実現される。

【０１４４】

または、ストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路１１により命令履歴テーブル１２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリからストア命令が実行されたか否かを示す値を読み出すのに加えて、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリに、ストア命令が実行されたことを示す値を格納することで、ストア命令間の出力依存関係が存在する可能性

を検出する機能を有するデータ依存関係検出装置が実現される。

【０１４５】

さらに、ロード命令が実行された場合には、該ロード命令の対象アドレスをハッシュ関数回路１１により命令履歴テーブル１２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリからストア命令が実行されたか否かを示す値を読み出すのに加えて、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリに、ロード命令が実行されたことを示す値を格納し、ストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路１１により命令履歴テーブル１２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリからロード命令が実行されたか否かを示す値及びストア命令が実行されたか否かを示す値を読み出すのに加えて、前記変換されたエントリ番号に対応する命令履歴テーブル１２のエントリに、ストア命令が実行されたことを示す値を格納することで、ストア命令からロード命令への正依存関係が存在する可能性、ロード命令からストア命令への逆依存関係が存在する可能性、及びストア命令間の出力依存関係が存在する可能性、のすべてを検出する機能を有するデータ依存関係検出装置が実現される。

【０１４６】

次に、本発明の第２の実施の形態におけるデータ依存関係検出装置について説明する。本発明の第２の実施の形態におけるデータ依存関係検出装置は、マルチプロセッサシステムにおいて、スレッド並列処理におけるデータ依存投機実行を実現する。ここでスレッド並列処理とは、単一のプログラムあるいは複数のプログラムをスレッドと呼ぶ複数のプログラムの部分単位に分割し、それぞれのスレッドを複数のプロセッサなどで並列に処理することでプログラム実行を高速化する、プログラム処理の高速化方法の一つである。

【０１４７】

スレッドは、プログラムの一部分単位であり、複数の命令からなる命令列なので、複数のスレッド間で順序関係が定義される場合がある。例えば、同一のプログラムに属する２つのスレッドをそれぞれＴ０、Ｔ１としたとき、Ｔ０とＴ１の

間には、そのプログラム中の位置関係により、明確に順序関係が存在する。

【０１４８】

仮に、プログラムにおいてＴ０がＴ１よりも前に位置しているならば、プログラムの意味上、Ｔ０がＴ１よりも前に実行されることを期待されていることになる。逆に、プログラムにおいてＴ１がＴ０よりも前に位置しているならば、プログラムの意味上、Ｔ１がＴ０よりも前に実行されることを期待されていることになる。しかし、Ｔ０及びＴ１に含まれる命令間に依存関係が存在しない場合、Ｔ０とＴ１間の順序関係を無視した順序で実行しても、プログラムの意味は変わらないため、正しいプログラムの実行結果を得ることができる。

【０１４９】

スレッド並列処理では、上記した性質を利用して、プログラムの正しい実行結果を保証しつつ、スレッド間の順序関係によらずにスレッドの実行を並列に行うことで、プログラム実行を高速化する。命令単位の非プログラム順序実行と対比して、スレッド単位の非プログラム順序実行と位置づけられる。

【０１５０】

スレッド単位の非プログラム順序実行においても、正しいプログラムの実行結果を得るためには、スレッドが含む命令間の依存関係の解消や保証が必要である。

【０１５１】

しかしながら、命令単位の非プログラム順序実行と同様に、特にメモリに関する正依存関係に対しては、本質的にプログラム順序で実行する必要があり、かつ、確定的にプログラム順序実行を行うと、非プログラム順序実行の実行性能向上の効果が十分得られないという問題がある。

【０１５２】

特に、スレッド単位の非プログラム順序実行では、複数の命令からなるスレッド単位で、非プログラム順序実行が妨げられるので、より問題は深刻である。

【０１５３】

この問題への対応としては、命令単位の非プログラム順序実行と同様に、データ依存投機実行が有効である。すわなち、スレッドが含む命令間の正依存関係の

有無が判明するよりも以前に、正依存関係が存在しないと仮定して、投機的にスレッドの非プログラム順序実行を行う、スレッド単位のデータ依存投機実行である。

【０１５４】

スレッド単位のデータ依存投機実行を実現するためには、順序関係が存在するスレッド対に関して、順序が前のスレッドが含むストア命令から、順序が後のスレッドが含むロード命令への正依存関係を検出する機能を必要とする。

【０１５５】

本発明の第２の実施の形態におけるデータ依存関係検出装置は、このスレッド間の正依存関係を検出する機能を有する。ここで、本発明の第２の実施の形態におけるデータ依存関係検出装置は、スレッド間での非プログラム順序実行は行うが、スレッド内のスレッドを構成する命令間では非プログラム順序実行を行わないとして、スレッド内の命令間の正依存検出の機能は備えない。

【０１５６】

次に図６を参照して、本発明の第２の実施の形態におけるデータ依存関係検出装置の構成を説明する。図６は、４つのプロセッサ、すなわちプロセッサ５０、プロセッサ５１、プロセッサ５２、プロセッサ５３、及びスレッド制御部５５から構成されるマルチプロセッサの構成を示すブロック図である。プロセッサ５０～５３はすべて同一の構成である。なお、図６において、プロセッサ５２及びプロセッサ５３の構成の詳細は省略しており、プロセッサ５２及びプロセッサ５３の同一の構成要素には同一の参照符号が付されている。

【０１５７】

スレッド制御部５５は、プロセッサ５０～５３の各々が実行するスレッドの割り当てや、スレッド間の順序関係の通知、スレッドのデータ依存投機実行が失敗した場合の回復処理の指示等、スレッド並列処理、及びデータ依存投機実行の制御を行う。プロセッサ５０～５３は、本発明の第２の実施の形態におけるデータ依存関係検出装置３０、プロセッサ制御部５７、及び命令実行部１８を含む。プロセッサの他の構成要素については図示していない。

【０１５８】

プログラムをプロセッサ５０～５３によってスレッド並列処理するにあたり、スレッド制御部５５は、プロセッサ５０～５３の各々が実行すべきスレッドの割り当てを行う。また、プロセッサ５０～５３のそれぞれに、そのスレッド実行はデータ依存投機実行か否かを示す投機実行フラグや他の３つのプロセッサが実行するスレッドとの順序関係を示すスレッド順序を出力する。

【０１５９】

プロセッサ５０～５３はそれぞれ、投機実行フラグ３９及びスレッド順序４０を入力する。

【０１６０】

スレッド制御部５５より割り当てられたスレッドを、各プロセッサが実行する場合、各プロセッサのプロセッサ制御部５７は、実行する命令の種別を、各プロセッサの命令実行部１８とデータ依存関係検出装置３０に出力するとともに、実行命令４６を通じて、他の３つのプロセッサに出力する。

【０１６１】

データ依存関係検出装置３０は、実行する命令種別を実行命令３４から入力する。同時に、データ依存関係検出装置３０は、他の３つのプロセッサが実行する命令種別を、実行命令３８を通じてそれぞれ入力する。さらに、実行する命令がロード命令またはストア命令の場合、プロセッサ制御部５７は、ロード命令またはストア命令の対象アドレスを、各プロセッサの命令実行部１８とデータ依存関係検出装置３０に出力するとともに、ロード・ストアアドレス４５を通じて、他の３つのプロセッサに出力する。データ依存関係検出装置３０は、ロード・ストア命令の対象アドレスをロード・ストアアドレス３５から入力する。同時に、他の３つのプロセッサが実行する命令がロード命令またはストア命令の場合、データ依存関係検出装置３０はその対象アドレスをロード・ストアアドレス３７を通じてそれぞれ入力する。

【０１６２】

例えば、プロセッサ５０がスレッド制御部５５より割り当てられたスレッドを実行する際、プロセッサ５０のプロセッサ制御部５７は実行する命令の種別をプロセッサ５０の命令実行部１８及びデータ依存関係検出装置３０に加えて、実行

命令４６を通じて他の３つのプロセッサ、プロセッサ５１、プロセッサ５２、プロセッサ５３に出力する。同時に、プロセッサ５０のデータ依存関係検出装置３０は他の３つのプロセッサ、プロセッサ５１、プロセッサ５２、プロセッサ５３が実行する命令種別を実行命令３８を通じてそれぞれ入力する。

【０１６３】

さらに、実行する命令がロード命令またはストア命令の場合、プロセッサ５０のプロセッサ制御部５７は、その対象アドレスをプロセッサ５０の命令実行部１８及びデータ依存関係検出装置３０、及びロード・ストアアドレス４５を通じて他の３つのプロセッサ、プロセッサ５１、プロセッサ５２、プロセッサ５３に出力する。同時に、他の３つのプロセッサ、プロセッサ５１、プロセッサ５２、プロセッサ５３が実行する命令がロード命令またはストア命令の場合、プロセッサ５０のデータ依存関係検出装置３０はその対象アドレスをロード・ストアアドレス３７を通じてそれぞれ入力する。

【０１６４】

プロセッサ５０～５３のそれぞれが備えるデータ依存関係検出装置３０は、４つのハッシュ関数回路３１、命令履歴テーブル３２及び論理和回路４１から構成される。また、自身が設置されているプロセッサ（「自プロセッサ」という）のプロセッサ制御部５７からの入力として、実行命令３４、ロード・ストアアドレス３５、及び、他の３つのプロセッサから、それぞれの実行命令３８とロード・ストアアドレス３７を入力する。また、スレッド制御部５５からの入力として、投機実行フラグ３９及びスレッド順序４０を備える。また、スレッド制御部５５への出力として、データ依存検出結果３６を備える。

【０１６５】

４つのハッシュ関数回路３１は、ロード命令またはストア命令の対象アドレスを、命令履歴テーブル３２のエントリ番号に変換するハッシュ関数ｆを実現する論理回路である。ここで、上記ハッシュ関数ｆは、同一の入力に対しては、同一の値をもつ。すなわち、Ｎ１＝ｆ（Ａ１）かつＮ２＝ｆ（Ａ２）としたとき、Ａ１＝Ａ２であればＮ１＝Ｎ２が成立する関数である。４つのハッシュ関数回路３１のうちの１つは、自プロセッサが実行するロード命令の対象アドレスを命令履

歴テーブル３２のエントリ番号に変換する。４つのハッシュ関数回路３１のうちの残りの３つはそれぞれ、他の３つのプロセッサが実行するストア命令の対象アドレスを命令履歴テーブル３２のエントリ番号に変換する。

【０１６６】

命令履歴テーブル３２は、１ビットの情報を格納できる複数のエントリから構成される。命令履歴テーブル３２が備えるエントリの総数は、ハッシュ関数回路３１の出力がｎビット幅である場合、２のｎ乗に等しい数である。命令履歴テーブル３２は、１つの書き込みポートと３つの読み出しポートを備える。すなわち、１つの書き込み処理と、３つの読み出し処理を同時に行うことができる。

【０１６７】

４つのハッシュ関数回路３１のうちの１つである、自プロセッサが実行するロード命令の対象アドレスを入力とするハッシュ関数回路３１の出力は、命令履歴テーブル３２の書き込みポートに接続され、ハッシュ関数回路３１が出力するエントリ番号が指し示すエントリに対して書き込み処理を行う。

【０１６８】

４つのハッシュ関数回路３１のうちの残りの３つである、他のプロセッサが実行するストア命令の対象アドレスを入力とするハッシュ関数回路３１の出力は、命令履歴テーブル３２の読み出しポートに接続され、ハッシュ関数回路３１が出力するエントリ番号が指し示すエントリに対して読み出し処理を行う。

【０１６９】

さらに、スレッド制御部５５からデータ依存関係検出装置３０に入力される投機実行フラグ３９に基づいて、すべてのエントリに同一の値を書き込む初期化機能を有する。

【０１７０】

論理和関数４１は、命令履歴テーブル３２が備える３つの読み出しポートの３つの読み出し結果に対して論理和を演算し、データ依存検出結果３６としてスレッド制御部５５に出力する。

【０１７１】

次に、図６を参照して、データ依存関係検出装置３０の動作について説明する

。まず、初期状態においては、命令履歴テーブル３２を構成するすべてのエントリは、自プロセッサがロード命令を実行していないことを示す第１の状態（例えば論理０）にある。自プロセッサがスレッド間のデータ依存関係に対して投機的な命令実行を行わず、プログラム順序で確定的に命令実行を行う状態にある場合、データ依存関係検出装置３０はなんら動作を行わない。

【０１７２】

それに対して、自プロセッサがスレッド間のデータ依存関係に対して投機的な命令実行を行う状態にある場合、データ依存関係検出装置３０は、他の３つのプロセッサのうち、自プロセッサが実行するスレッドよりもプログラム順序で前の順序にあるスレッド（これを「先行するスレッド」という）を割り当てられたプロセッサが実行するストア命令から、自プロセッサが実行するロード命令への正依存関係の検出処理を行う。

【０１７３】

自プロセッサが確定実行状態または投機実行状態のどちらにあるかは、スレッド制御部５５からデータ依存関係検出装置３０に入力される投機実行フラグ３９で判断する。

【０１７４】

また、自プロセッサが実行するスレッドが、他の３つのプロセッサが実行するそれぞれのスレッドよりもプログラム順序で前の順序にある否かは、スレッド制御部５５からデータ依存関係検出装置３０に入力されるスレッド順序４０で判断する。また、他の３つのプロセッサが実行している命令の種別は、他の３つのプロセッサからデータ依存関係検出装置３０に入力される実行命令３８で判断する。

【０１７５】

自プロセッサが投機実行状態にあるときに、投機的なロード命令が実行されると、データ依存関係検出装置３０は、投機的なロード命令が実行されたことを命令履歴テーブル３２に記憶する。より詳細には、ロード命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、命令履歴テーブル３２の書き込みポートに入力する。命令履歴テーブル３２は、入力

されたエントリ番号に対応するエントリを、投機的なロード命令が実行されたことを示す第２の状態（例えば論理１）に変更する。

【０１７６】

一方、自プロセッサが投機実行状態にあるときに、他の３つのプロセッサのうち、自プロセッサが実行するスレッドよりもプログラム順序で前の順序にあるスレッド、すなわち先行するスレッドを割り当てられたプロセッサでストア命令が実行された場合には、データ依存関係検出装置３０は、対応する命令履歴テーブル３２のエントリに記憶された状態の読み出しを行う。より詳細には、ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、命令履歴テーブル３２の３つの入力ポートのうち対応するものに入力する。命令履歴テーブル３２は、入力されたエントリ番号に対応するエントリの内容を読み出し、論理和回路４１に出力する。論理和回路４１は、命令履歴テーブル３２の３つの読み出しポートの出力の論理和を演算し、データ依存検出結果３６としてスレッド制御部５５に出力する。

【０１７７】

他の３つのプロセッサのうち、自プロセッサが実行するスレッドよりも先行するスレッドを実行するプロセッサのいずれかが実行したストア命令の対象アドレスが、自プロセッサがそれまでデータ依存投機実行したロード命令の対象アドレスと等しいか、あるいは等しくない場合でも、エイリアスが発生して、命令履歴テーブル３２の同じエントリに割り当てられた場合、命令履歴テーブル３２からデータ依存投機実行されたロード命令が存在することを示す値が読み出されるため、正依存関係が存在する可能性があることが検出される。この場合、データ依存関係検出装置３０は、データ依存投機実行が失敗したことを示す値（例えば論理１）をデータ依存検出結果３６を通じてスレッド制御部５５に出力する。

【０１７８】

スレッド制御部５５は、プロセッサ５０～５３のいずれかより、データ依存投機実行の失敗の通知を受け取ると、その失敗通知を出力したプロセッサ及び、その失敗通知を出力したプロセッサが実行していたスレッドよりも、プログラム順序で後に位置するスレッドを実行しているプロセッサに対して、データ依存投機

実行の失敗による回復処理の実行要求を出力する。

【０１７９】

データ依存投機実行の失敗による回復処理の実行要求の出力対象となった各プロセッサでは、データ依存投機実行の失敗による回復処理の実行要求が、回復処理実行要求４７を通じてプロセッサ制御部５７に通知される。

【０１８０】

データ依存投機実行の失敗を通知したプロセッサだけではなく、データ依存投機実行を失敗したスレッドよりプログラム順序が後のスレッドを実行しているプロセッサに対しても、データ依存投機実行の失敗による回復処理の実行を要求するのは、次に示す理由による。すなわち、データ依存投機実行の失敗を通知をしたプロセッサは、正依存関係に違反してロード命令をデータ依存投機実行した可能性がある。そのため、前記データ依存投機実行の失敗を通知をしたプロセッサが実行していた、データ依存投機実行に失敗したスレッドの実行結果は正しくない可能性がある。従って、前記データ依存投機実行に失敗したスレッドの実行結果を参照している可能性のある、それよりもプログラム順序で後に位置するスレッドの実行結果もまた、正しくない可能性がある。

【０１８１】

同様にして、データ依存投機実行に失敗したスレッドの実行結果を参照している可能性のある前記スレッドの実行結果を参照している可能性のある、それよりもさらにプログラム順序で後に位置するスレッドの実行結果もまた、正しくない可能性がある。すなわち、データ依存投機実行の失敗を通知をしたプロセッサが実行していたスレッドよりもプログラム順序が後に位置するすべてのスレッドの実行結果は正しくない可能性がある。このため、データ依存投機実行の失敗を通知したプロセッサだけではなく、データ依存投機実行を失敗したスレッドよりプログラム順序が後のスレッドを実行しているプロセッサに対しても、データ依存投機実行の失敗による回復処理の実行を要求し、スレッドの実行結果の正しさを保証する。

【０１８２】

一方、データ依存投機実行の失敗により、スレッド制御部５５からのデータ依

存投機実行の失敗による回復処理の実行要求の対象になったプロセッサでは、データ依存投機実行の失敗による回復処理の実行要求が、各プロセッサの回復処理実行要求４７を通じてプロセッサ制御部５７に通知される。その場合、上記各プロセッサは、データ依存投機実行を行ったスレッドの再実行を行うなど、データ依存投機実行の失敗による回復処理を実行することで、スレッドの実行結果の正しさを保証する。

【０１８３】

上述したように、本発明の第２の実施の形態によるデータ依存関係検出装置３０は、自プロセッサが実行するスレッドよりプログラム順序で先行するスレッドに含まれるストア命令から、自プロセッサが実行するスレッドに含まれるロード命令への正依存関係を検出する機能を有する。この機能は、データ依存関係検出装置３０が備える命令履歴テーブル３２が、自プロセッサがロード命令をデータ依存投機実行する際に書き込みを行う書き込みポートを備えると共に、他のプロセッサがストア命令を実行する際に読み出しを行う複数の読み出しポートを備えることで実現される。この機能によりスレッド並列処理において、スレッド単位のデータ依存投機実行の実現を可能とすることで、スレッド単位の並列実行によるプログラム実行のさらなる高速化を可能としている。

【０１８４】

また、本発明の第２の実施の形態によるデータ依存関係検出装置３０は、前述した本発明の第１の実施の形態によるデータ依存関係検出装置１０と同様に、ストア命令からロード命令への正依存関係の存在を厳密に検出するのではなく、ストア命令からロード命令への正依存関係の存在の可能性を検出するものである。任意のロード命令の対象アドレスは、ハッシュ関数回路３１により、命令履歴テーブル３２の特定のエントリに重複を許して割り当てるため、従来のデータ依存関係検出装置と異なり、実行されたロード命令の対象アドレス自体をすべて記憶しておく必要がなく、またストア命令の対象アドレスと比較するためのアドレス比較器も必要としない。

【０１８５】

すなわち、本発明の第２の実施の形態のデータ依存関係検出装置によれば、デ

ータ依存投機実行の成否判定に必要なハードウエア量が小さいという効果を得ることができる。

【０１８６】

さらに、本発明の第２の実施の形態のデータ依存関係検出装置によれば、データ依存投機実行することのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもデータ依存投機実行による実行性能の向上を享受することのできる。特に、スレッド単位のデータ依存投機実行では、命令単位のデータ依存投機実行に比べてデータ依存投機実行されるロード・ストア命令数が多いため、データ依存投機実行することのできるロード命令の数がデータ依存関係検出装置のハードウエア量に制限されない、という本発明の特徴は、プログラム実行性能の向上に大きな効果を奏する。

【０１８７】

次に、図６及び図７を参照して、本発明の第２の実施の形態によるスレッド並列処理におけるデータ依存投機実行の具体例を以下に示す。まず、図７（ａ）は、ロード命令ＬＤ１、ＬＤ２、ＬＤ３、ＬＤ４、ＬＤ５、ストア命令ＳＴ１、ＳＴ２、ＳＴ３、及びロード・ストア命令以外の命令を含む１２命令からなるプログラム例における命令のプログラム順序である。図中、空欄の四角はロード・ストア命令以外の命令を示す。

【０１８８】

プログラム順序は順に、ロード命令ＬＤ１、ロード・ストア命令以外の命令、ストア命令ＳＴ１、ロード命令ＬＤ２、ストア命令ＳＴ２、ロード・ストア命令以外の命令、ロード命令ＬＤ３、ロード・ストア命令以外の命令、ロード命令ＬＤ４、ロード命令ＬＤ５、ストア命令ＳＴ３、ロード・ストア命令以外の命令、の順序である。ここで、ＬＤ１の対象アドレスはＡ１、ＬＤ２の対象アドレスはＡ２、ＬＤ３の対象アドレスはＡ３、ＬＤ４の対象アドレスはＡ４、ＬＤ５の対象アドレスはＡ５、ＳＴ１の対象アドレスはＡ３、ＳＴ２の対象アドレスはＡ１、ＳＴ３の対象アドレスはＡ３、である。

【０１８９】

次に、図７（ａ）に示したプロクラム順序のプログラム例を、図６に示した４

つのプロセッサ、プロセッサ５０、プロセッサ５１、プロセッサ５２、プロセッサ５３、においてスレッド単位でデータ依存投機実行を行うとする。具体的には、図７（ａ）に示した１２命令から構成されるプログラム例を３命令から構成される４つのスレッドに分割し、それぞれを４つのプロセッサで並列実行を行う。すなわち、プログラム順序で順に、ロード命令ＬＤ１、ロード・ストア命令以外の命令、ストア命令ＳＴ１、の３命令列をスレッド０、ロード命令ＬＤ２、ストア命令ＳＴ２、ロード・ストア命令以外の命令、の３命令列をスレッド１、ロード命令ＬＤ３、ロード・ストア命令以外の命令、ロード命令ＬＤ４、の３命令列をスレッド２、ロード命令ＬＤ５、ストア命令ＳＴ３、ロード・ストア命令以外の命令、の３命令列をスレッド３として、スレッド０をプロセッサ５２で、スレッド１をプロセッサ５１で、スレッド２をプロセッサ５０で、スレッド３をプロセッサ５３で並列に実行するとする。この各プロセッサが実行するスレッドの割り当ては、スレッド制御部５５によって行われる。

【０１９０】

ＬＤ３とＳＴ１の対象アドレスは同じＡ３であり、プログラム順序でＳＴ１の方が前にあるため、ＳＴ１からＬＤ３にはデータ依存関係が存在する。すなわち、実行時においてＬＤ３よりもＳＴ１が前に実行されて、ＳＴ１がアドレスＡ３に書き込みを行った結果を、ＬＤ３がアドレスＡ３から読み出しを行わなければ、プログラムの実行結果の正しさは保証されない。スレッド単位の並列処理においては、ＳＴ１を含むスレッド０からＬＤ３を含むスレッド２へ正依存関係が存在することになる。しかしながら、通常、各プロセッサへのスレッドを割り当てる時点では、ＳＴ１からＬＤ３への正依存関係、すなわちスレッド０からスレッド２への存在は判明しない。このため、確定的にスレッドの並列実行を行うとすると、正依存関係が無いことが判明していないスレッド間では、プログラム順序でスレッドを逐次に実行することで、正依存関係を保証する必要があるため、スレッドの並列処理によるプログラム実行の高速性を充分得ることができない。

【０１９１】

一方、スレッド単位のデータ依存投機実行では、スレッド間の正依存関係の存在の有無が判明していなくても、投機的にスレッドの並列実行を行う。データ依

存投機実行が成功する確率が充分高ければ、スレッドの並列処理によるプログラム実行の高速化が達成される。図７（ｂ）は、図７（ａ）に示したプログラム順序の命令列、スレッド列のデータ依存投機実行によるスレッド並列実行の例であり、左から順にサイクル番号、スレッド２を実行するプロセッサ５０のそのサイクルの実行命令、その実行命令の対象アドレス、スレッド１を実行するプロセッサ５１のそのサイクルの実行命令、その実行命令の対象アドレス、スレッド０を実行するプロセッサ５２のそのサイクルの実行命令、その実行命令の対象アドレス、スレッド３を実行するプロセッサ５３のそのサイクルの実行命令、その実行命令の対象アドレス、をそれぞれ示す。すなわち、スレッド２を実行するプロセッサ５０では、１サイクル目にＬＤ３、２サイクル目にロード・ストア命令以外の命令、３サイクル目にＬＤ４、が実行される。スレッド１を実行するプロセッサ５１では、１サイクル目にＬＤ２、２サイクル目にＳＴ２、３サイクル目にロード・ストア命令以外の命令、が実行される。スレッド０を実行するプロセッサ５２では、１サイクル目にＬＤ１、２サイクル目にロード・ストア命令以外の命令、３サイクル目にＳＴ１、が実行される。スレッド３を実行するプロセッサ５３では、１サイクル目にＬＤ５、２サイクル目にＳＴ３、３サイクル目にロード・ストア命令以外の命令、が実行される。

【０１９２】

ここで、スレッド０はプログラム順序で最も前に位置するスレッドのため、スレッド１～３に含まれるストア命令からスレッド０に含まれるロード命令への正依存関係は存在しない。すなわち、プロセッサ５２は確定的にスレッド０を実行することができる。一方、スレッド１～３に関しては、正依存関係の有無が不明であるため、プロセッサ５０、プロセッサ５１、及びプロセッサ５３はスレッド１～３をデータ依存投機実行することになる。図７（ｂ）に示した実行例では、仮にデータ依存投機実行が成功した場合、１２命令からなるプログラムが３サイクルで実行されることになり、大幅なプログラム実行性能の向上が達成される。

【０１９３】

しかしながら、図７（ｂ）に示した実行例の場合、正依存関係のあるＳＴ１及びＬＤ３について、ＬＤ３が１サイクル目にプロセッサ５０で実行され、ＳＴ１

が３サイクル目にプロセッサ５２で実行されるため、正依存関係が保証されておらず、正しいプログラムの実行結果が得られない。すなわち、プロセッサ５０におけるスレッド２のデータ依存投機実行は失敗している。プロセッサ５０が備えるデータ依存関係検出装置３０は、プロセッサ５２が実行したＳＴ１からプロセッサ５０が実行したＬＤ３への正依存関係の存在を検出し、スレッド２のデータ依存投機実行が失敗したことをスレッド制御部５５に通知する機能を担う。スレッド制御部５５は、プロセッサ５０よりデータ依存投機実行が失敗した旨を通知されると、正しいプログラムの実行結果が得られるように、プロセッサ５０及びデータ依存投機実行が失敗したスレッド２よりプログラム順序で後に位置するスレッド３を実行するプロセッサ５３に対して、スレッドの実行結果の取り消しやスレッドの再実行など、データ依存投機実行失敗による回復処理を要求する。

【０１９４】

図７（ｃ）は、図７（ｂ）に示したプロセッサ５０の命令実行順序における、プロセッサ５０が備えるデータ依存関係検出装置３０の動作を示す図である。図７（ｃ）に左から順にサイクル番号、そのサイクルの命令履歴テーブル３２の状態をそれぞれ示す。命令履歴テーブル３２は４つのエントリから構成されるとして、４つのエントリの状態を右から順に、エントリ００、エントリ０１、エントリ１０、エントリ１１、の内容を示している。

【０１９５】

０サイクル目では、命令履歴テーブル３２は初期状態にあり、すべてのエントリの内容はロード命令が実行されていないことを示す論理０である。次に、１サイクル目ではプロセッサ５０において、スレッド２に含まれるロード命令ＬＤ３がデータ依存投機実行される。ＬＤ３の対象アドレスＡ３がハッシュ関数回路３１により、命令履歴テーブル３２のエントリ番号００に変換されるとすると、命令履歴テーブル３２のエントリ００の内容をロード命令が実行されたことを示す論理１にする。

【０１９６】

次に、２サイクル目ではプロセッサ５１においてスレッド１に含まれるＳＴ２が、プロセッサ５３においてスレッド３に含まれるＳＴ３が実行される。プロセ

ッサ５１が実行しているスレッド１は、プロセッサ５０がデータ依存投機実行しているスレッド２よりもプログラム順序で前に位置するため、正依存検出の対象となる。他のプロセッサが実行している命令種別は実行命令３８で、その実行命令の対象アドレスはロード・ストアアドレス３７で、データ依存関係検出装置３０に与えられている。ＳＴ２の対象アドレスＡ２がハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号１０に変換されるとすると、データ依存関係検出装置３０は命令履歴テーブル３２のエントリ１０の内容を読み出して、ＳＴ２から自プロセッサ（プロセッサ５０）が実行したロード命令への正依存関係の存在の有無を判定する。この場合、命令履歴テーブル３２のエントリ１０からは、ロード命令が実行されていないことを示す論理０が読み出されるため、プロセッサ５０におけるスレッド２のデータ依存投機実行は、プロセッサ５１が実行しているスレッド１に対して成功していると判断される。

【０１９７】

一方、プロセッサ５３が実行しているスレッド３は、プロセッサ５０がデータ依存投機実行しているスレッド２よりもプログラム順序で後に位置するため、正依存検出の対象とならない。すなわち、スレッド３に含まれるＳＴ３の対象アドレスＡ３は、プロセッサ５０が１サイクル目に実行したＬＤ３の対象アドレスと同じであるが、プログラム順序はＬＤ３が前、ＳＴ３が後であるため、正依存関係は常に存在しない。このため、プロセッサ５０が備える命令履歴テーブル３２は、プロセッサ５３が実行するスレッド３に含まれるストア命令に対しては、データ依存投機実行の成否判定を行わない。結局、２サイクル目において、プロセッサ５０が備える命令履歴テーブル３２は、データ依存投機実行が成功したとして、そのままスレッド実行を続行する。

【０１９８】

次に、３サイクル目ではプロセッサ５０においてスレッド２に含まれるＬＤ４が、プロセッサ５２においてスレッド０に含まれるＳＴ１が実行される。まず、ＬＤ４に対してプロセッサ５０が備える命令履歴テーブル３２は、ＬＤ４の対象アドレスＡ４がハッシュ関数回路３１により、命令履歴テーブル３２のエントリ番号１１に変換されるとすると、命令履歴テーブル３２のエントリ１の内容をロ

ード命令が実行されたことを示す論理１にする。一方、プロセッサ５２が実行しているスレッド０は、プロセッサ５０がデータ依存投機実行しているスレッド２よりもプログラム順序で前に位置し、正依存検出の対象となるため、プロセッサ５２が実行するＳＴ１に対しては、自プロセッサ（プロセッサ５０）が実行したロード命令への正依存関係の存在の有無を判定する。具体的には、ＳＴ１の対象アドレスＡ３はハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号００に変換されるので、データ依存関係検出装置３０は命令履歴テーブル３２のエントリ００の内容を読み出して、ＳＴ１から自プロセッサ（プロセッサ５０）が実行したロード命令への正依存関係の存在の有無を判定する。この場合、命令履歴テーブル３２のエントリ００からは、０サイクル目にＬＤ３により格納された論理１が読み出される。すなわち、ＳＴ１からＬＤ３への正依存関係が検出されるため、プロセッサ５０におけるスレッド２のデータ依存投機実行は、プロセッサ５２が実行しているスレッド０に対して失敗したと判断され、スレッド制御部５５にその旨がデータ依存検出結果３６を通じて通知される。

【０１９９】

スレッド制御部５５は、プロセッサ５０よりデータ依存投機実行が失敗した旨を通知されると、正しいプログラムの実行結果が得られるように、プロセッサ５０及びデータ依存投機実行が失敗したスレッド２よりプログラム順序で後に位置するスレッド３を実行するプロセッサ５３に対して、スレッドの実行結果の取り消し及び、スレッド再実行を要求し、データ依存投機実行失敗の回復を図る。

【０２００】

以上、プロセッサ５０について、データ依存投機実行における、データ依存関係検出装置３０による正依存関係の検出及びデータ依存投機実行の正否判定の動作について説明したが、スレッド１を実行するプロセッサ５１、スレッド０を実行するプロセッサ５２、及びスレッド３を実行するプロセッサ５３も同様に、それぞれが備えるデータ依存関係検出装置３０により、正依存関係の検出及びデータ依存投機実行の正否判定が行われる。プロセッサ５１及びプロセッサ５２においては、正依存関係が検出されず、データ依存投機実行が成功したと判断されるため、３サイクル目で実行が完了する。一方、プロセッサ５３においても正依存

関係が検出されず、データ依存投機実行が成功したと判断されるが、プロセッサ５３が実行するスレッド３よりもプログラム順序で前に位置するスレッド２を実行するプロセッサ５０がデータ依存投機実行を失敗したため、データ依存投機実行失敗による回復処理が要求される。

【０２０１】

スレッド制御部５５よりデータ依存投機実行失敗による回復処理が要求されるプロセッサ５０及びプロセッサ５３は、回復処理実行要求４７を通じてそれぞれのプロセッサ制御部５７がその要求を受け取ると、まず３サイクル目までのスレッドの実行結果の取り消しを行う。次に、５サイクル目からスレッドの再実行を開始する。スレッド０及びスレッド１が既に実行を完了しているため、スレッド２は実行が完了していないスレッド、すなわちスレッド２及びスレッド３のうち、プログラム順序で一番前にあるスレッドであり、従って確定的に実行することができる。具体的には、プロセッサ５０は、５サイクル目にＬＤ３、６サイクル目にロード・ストア命令以外の命令、７サイクル目にＬＤ４、を確定的に再実行する。５サイクル目に実行するＬＤ３に対しては、スレッド０に含まれるＳＴ１から正依存関係が存在するが、ＳＴ１は既にプロセッサ５２により３サイクル目に実行されているため、正依存関係に違反しない。一方、スレッド３は、スレッド２が含むストア命令から、スレッド３が含むロード命令への正依存関係、すなわち、スレッド２からの正依存関係の存在が不明であるため、再びデータ依存投機実行されることになる。具体的には、プロセッサ５３は、５サイクル目にＬＤ５、６サイクル目にＳＴ３、７サイクル目にロード・ストア命令以外の命令、を投機的に再実行する。この場合、スレッド２からの正依存関係は検出されないため、データ依存投機実行は成功したとして、７サイクル目でスレッド３の実行は完了する。

【０２０２】

以上、本発明の第２の実施の形態によるデータ依存投機実行の具体例を説明した。１２命令からなるプログラム例に対して、４つのプロセッサによりスレッド並列処理を行うと、データ依存投機実行が成功した場合は３サイクルで、データ依存投機実行が１回失敗した場合でも７サイクルで実行が完了し、プログラムの

実行性能の大幅は向上を望むことが可能となる。上記に示した具体例では、説明のため各スレッドを３命令から構成したが、通常は数十命令から数万命令で構成されるので、より多くのロード命令がデータ依存投機実行されることになる。しかしながら、ロード命令の対象アドレスをすべて記憶する必要のある従来のデータ依存関係検出装置１００では、アドレスバッファ１０１に空きがなくなるとデータ依存投機実行することができないため、それ以降は逐次的かつ確定的にスレッド実行をせざるおえない。すなわち、従来のデータ依存関係検出装置１００では、スレッド並列処理におけるデータ依存投機実行によるプログラム実行の高速化の効果を充分に享受することができない、あるいは、享受するためには膨大なハードウエア量が必要となるという問題がある。

【０２０３】

一方、本発明の第２の実施の形態によるデータ依存関係検出装置３０は、ロードストア命令の対象アドレスに関してエイリアスの存在を許容することで、ロード命令の対象アドレスそのものをすべて記憶する必要をなくしたことを特徴としている。この特徴により、データ依存投機実行が可能なロード命令数に制限がなく、必要なハードウエア量も小さい、という効果を得ることができる。また、正依存関係の検出の処理時間が短いため、プロセッサの動作周波数を高速化することが容易であるという効果を得ることができる。さらに、データ依存関係検出装置３０は、スレッド制御部５５が出力するスレッド順序４０に基づいて、正依存検出する対象のスレッドを選択する機能を有することを特徴としている。この特徴により、スレッド間の順序に柔軟に対応して、スレッド間の正依存関係を適切に検出できる、という効果を得ることができる。

【０２０４】

上記したデータ依存関係検出装置３０では、スレッド制御部５５が出力するスレッド順序４０を参照して、自プロセッサよりも先行するスレッドを実行している他のプロセッサを特定し、それら先行するスレッドを実行するプロセッサがストア命令を実行するときのみ、命令履歴テーブル３２の読み出しを行い、次に論理和関数４１でそれら読み出し結果の論理和を演算することで、先行するスレッドから自プロセッサが実行するスレッドへの正依存関係の存在の検出を実現した

。一方、他の実現方法として、自プロセッサよりも先行するしかないにもかかわらず、他のプロセッサがストア命令を実行した場合には、常に命令履歴テーブル３２の読み出しを行っても良い。この場合、論理和関数４１でそれら読み出し結果の論理和を演算する際に、スレッド制御部５５が出力するスレッド順序４０を参照して、自プロセッサよりも先行するスレッドを実行している他のプロセッサを特定し、それら先行するスレッドによる読み出した結果のみ論理和を演算することで、先行するスレッドから自プロセッサが実行するスレッドへの正依存関係の存在の検出を実現することができる。

【０２０５】

また、上記した本発明の第２の実施の形態は、スレッド並列処理におけるスレッド単位のデータ依存投機実行において、自プロセッサでロード命令が実行された場合には、該ロード命令が処理対象とするアドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリに、ロード命令が実行されたことを示す値を格納し、他のプロセッサでストア命令が実行された場合には、該ストア命令が処理対象とするアドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリから、自プロセッサでロード命令が実行されたか否かを示す値を読み出すことで、他のプロセッサで実行されたストア命令から、自プロセッサで実行されたロード命令への正依存関係が存在する可能性を検出することを特徴とした。しかしながら、この実施の形態は、スレッド間の正依存関係が存在する可能性の検出にとどまらず、スレッド間で逆依存関係や出力依存関係が存在する可能性の検出に対しても、同様の構成で容易に適用することができる。

【０２０６】

例えば、自プロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリに、自プロセッサでストア命令が実行されたことを示す値を格納し、他のプロセッサでロード命令が実行された場合には、該ロード命令の対象アドレスをハ

ッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリから、自プロセッサでストア命令が実行されたか否かを示す値を読み出すことで、他のプロセッサで実行されたロード命令から、自プロセッサで実行されたストア命令への逆依存関係が存在する可能性を検出する機能を有するデータ依存関係検出装置が実現される。

【０２０７】

または、他のプロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリから、自プロセッサでストア命令が実行されたか否かを示す値を読み出し、自プロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリに、自プロセッサでストア命令が実行されたことを示す値を格納することで、他のプロセッサで実行されたストア命令から、自プロセッサで実行されたストア命令への出力依存関係が存在する可能性を検出する機能を有するデータ依存関係検出装置が実現される。

【０２０８】

さらに、他のプロセッサでロード命令が実行された場合には、該ロード命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリから、自プロセッサでストア命令が実行されたか否かを示す値を読み出し、他のプロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリから、自プロセッサでロード命令が実行されたか否かを示す値及び自プロセッサでストア命令が実行されたか否かを示す値を読み出し、自プロセッサでロード命令が実行された場合には、該ロード命令の対象アドレスをハッシュ関数回路３１に

より命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリに、自プロセッサでロード命令が実行されたことを示す値を格納し、自プロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル３２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル３２のエントリに、自プロセッサでストア命令が実行されたことを示す値を格納することで、他プロセッサで実行されたストア命令から、自プロセッサで実行されたロード命令への正依存関係が存在する可能性、及び他プロセッサで実行されたロード命令から、自プロセッサで実行されたストア命令への逆依存関係が存在する可能性、及び他のプロセッサで実行されたストア命令から、自プロセッサで実行されたストア命令への出力依存関係が存在する可能性、のすべてを検出する機能を有するデータ依存関係検出装置が実現される。

【０２０９】

また、上記した本発明の第２の実施の形態によるデータ依存関係検出装置３０は、スレッド間の正依存関係を検出する機能は有するが、スレッド内の命令間の正依存関係を検出する機能は有しない。すなわち、スレッド並列処理におけるスレッド単位のデータ依存投機実行は可能であるが、スレッド内の命令単位のデータ依存投機実行は行うことができない。そこで、スレッド間の正依存関係の検出機能のみならず、スレッド内の命令間の正依存関係の検出機能も備えることで、スレッド単位のデータ依存投機実行を実現するのに加えて、命令単位のデータ依存投機実行を可能とした、データ依存関係検出装置の第３の実施の形態を次に示す。

【０２１０】

図８を参照して、本発明の第３の実施の形態におけるデータ依存関係検出装置６０の構成を説明する。図８は、本発明の第３の実施の形態におけるデータ依存関係検出装置６０を備え、プロセッサ制御部６３、及び命令実行部１８を含むプロセッサ５９の構成を示すブロック図である。プロセッサの他の構成要素や、他のプロセッサ、スレッド制御部などは図示していない。プロセッサ５９の大部分は、本発明の第２の実施の形態によるデータ依存関係検出装置３０を備えるプロ

セッサ５０～５３と同様である。

【０２１１】

スレッド制御部（図示せず）より割り当てられたスレッドを、プロセッサ５９が実行する場合、プロセッサ５９が備えるプロセッサ制御部６３は、スレッドを構成する命令を、プログラム順序で確定的に実行するか、非プログラム順序でデータ依存投機実行を行うか、を示す投機実行フラグ３３をデータ依存関係検出装置６０に出力する。また、プロセッサ制御部６３は、実行する命令の種別を命令実行部１８及びデータ依存関係検出装置６０、及び実行命令４６を通じて他の３つのプロセッサに出力する。データ依存関係検出装置６０はその実行する命令種別を実行命令３４から入力する。同時に、データ依存関係検出装置６０は他の３つのプロセッサが実行する命令種別を実行命令３８を通じてそれぞれ入力する。さらに、実行する命令がロード命令またはストア命令の場合、プロセッサ制御部６３は、その対象アドレスを命令実行部１８及びデータ依存関係検出装置６０、及びロード・ストアアドレス４５を通じて他の３つのプロセッサに出力する。データ依存関係検出装置６０はそのロード・ストア命令の対象アドレスをロード・ストアアドレス３５から入力する。同時に、他の３つのプロセッサが実行する命令がロード命令またはストア命令の場合、データ依存関係検出装置６０はその対象アドレスをロード・ストアアドレス３７を通じてそれぞれ入力する。

【０２１２】

データ依存関係検出装置６０は、４つのハッシュ関数回路３１、命令履歴テーブル６２及び論理和回路６１から構成される。また、自プロセッサのプロセッサ制御部６３からの入力として、投機実行フラグ３３、実行命令３４、ロード・ストアアドレス３５及び、他の３つのプロセッサから、それぞれの実行命令３８とロード・ストアアドレス３７を入力する。また、スレッド制御部からの入力として、投機実行フラグ３９及びスレッド順序４０を備える。また、スレッド制御部への出力として、データ依存検出結果６４を備える。

【０２１３】

４つのハッシュ関数回路３１は、ロード命令またはストア命令の対象アドレスを、命令履歴テーブル６２のエントリ番号に変換するハッシュ関数ｆを実現する

論理回路である。ここで、上記ハッシュ関数ｆは、同一の入力に対しては、同一の値をもつことを特徴とする。すなわち、Ｎ１＝ｆ（Ａ１）かつＮ２＝ｆ（Ａ２）としたとき、Ａ１＝Ａ２であればＮ１＝Ｎ２が成立する関数である。４つのハッシュ関数回路３１のうちの１つは、自プロセッサが実行するロード命令及びストア命令の対象アドレスを命令履歴テーブル６２のエントリ番号に変換する。４つのハッシュ関数回路３１のうちの残りの３つはそれぞれ、他の３つのプロセッサが実行するストア命令の対象アドレスを命令履歴テーブル６２のエントリ番号に変換する。

【０２１４】

命令履歴テーブル６２は、１ビットの情報を格納できる複数のエントリから構成される。備える上記エントリの総数は、ハッシュ関数回路３１の出力がｎビット幅である場合、２のｎ乗に等しい数である。命令履歴テーブル６２は、１つの読み出し書き込みポートと３つの読み出しポートを備えることを特徴とする。すなわち、１つの読み出し処理または書き込み処理と、３つの読み出し処理を同時に行うことができる。４つのハッシュ関数回路３１のうちの１つである、自プロセッサが実行するロード命令またはストア命令の対象アドレスを入力とするハッシュ関数回路３１の出力は、命令履歴テーブル６２の読み出し書き込みポートに接続され、ハッシュ関数回路３１が出力するエントリ番号が指し示すエントリに対して読み出し処理または書き込み処理を行う。４つのハッシュ関数回路３１のうちの残りの３つである、他のプロセッサが実行するストア命令の対象アドレスを入力とするハッシュ関数回路３１の出力は、命令履歴テーブル６２の読み出しポートに接続され、ハッシュ関数回路３１が出力するエントリ番号が指し示すエントリに対して読み出し処理を行う。さらに、スレッド制御部からデータ依存関係検出装置６０に入力される投機実行フラグ３９及び、プロセッサ制御部６３からデータ依存関係検出装置６０に入力される投機実行フラグ３３に基づいて、すべてのエントリに同一の値を書き込む初期化機能を有する。論理和関数４１は、命令履歴テーブル６２が備える１つの読み出し書き込みポート及び３つの読み出しポートの合計４つの読み出し結果に対して論理和を演算し、データ依存検出結果６４としてスレッド制御部に出力する。

【０２１５】

次に、上記構成のデータ依存関係検出装置６０の動作を説明する。まず、初期状態においては、命令履歴テーブル６２を構成するすべてのエントリは、自プロセッサがロード命令を実行していないことを示す第１の状態（例えば論理０）にある。自プロセッサがスレッド間のデータ依存関係に対して投機的なスレッド実行を行わず、プログラム順序で確定的にスレッド実行を行う状態にある場合であり、かつ自プロセッサがスレッド内の命令間のデータ依存関係に対して投機的な命令実行を行わず、プログラム順序で確定的に命令実行を行う状態にある場合、データ依存関係検出装置６０はなんら動作を行わない。

【０２１６】

それに対して、自プロセッサがスレッド間のデータ依存関係に対して投機的なスレッド実行を行う状態にある場合、データ依存関係検出装置６０は、他の３つのプロセッサのうち、自プロセッサが実行するスレッドよりもプログラム順序で前の順序にあるスレッドを割り当てられたプロセッサが実行するストア命令から、自プロセッサが実行するロード命令への正依存関係の検出処理を行う。自プロセッサがスレッド実行に関して確定実行状態または投機実行状態のどちらにあるかは、スレッド制御部からデータ依存関係検出装置６０に入力される投機実行フラグ３９で判断する。また、自プロセッサが実行するスレッドが、他の３つのプロセッサが実行するそれぞれのスレッドよりもプログラム順序で前の順序にある否かは、スレッド制御部からデータ依存関係検出装置６０に入力されるスレッド順序４０で判断する。また、他の３つのプロセッサが実行している命令の種別は、他の３つのプロセッサからデータ依存関係検出装置６０に入力される実行命令３８で判断する。

【０２１７】

また、自プロセッサがスレッド内の命令間のデータ依存関係に対して投機的な命令実行を行う状態にある場合、データ依存関係検出装置６０は、自プロセッサがそれ以前に実行したストア命令から、自プロセッサが実行するロード命令への正依存関係の検出処理を行う。自プロセッサがスレッド内の命令の実行に関して確定実行状態または投機実行状態のどちらにあるかは、プロセッサ制御部６３か

らデータ依存関係検出装置６０に入力される投機実行フラグ３３で判断する。自プロセッサが実行する命令の種別は、プロセッサ制御部６３からデータ依存関係検出装置６０に入力される実行命令３４で判断する。

【０２１８】

自プロセッサがスレッド実行に関して、あるいはスレッド内の命令実行に関して投機実行状態にあるときに、投機的なロード命令が実行されると、データ依存関係検出装置６０は、投機的なロード命令が実行されたことを命令履歴テーブル６２に記憶する。より詳細には、ロード命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、命令履歴テーブル６２の読み出し書き込みポートに入力する。命令履歴テーブル６２は、入力されたエントリ番号に対応するエントリを、投機的なロード命令が実行されたことを示す第２の状態（例えば論理１）に変更する。

【０２１９】

一方、自プロセッサがスレッド実行に関して投機実行状態にあるときに、他の３つのプロセッサのうち、自プロセッサが実行するスレッドよりもプログラム順序で前の順序にあるスレッド、すなわち先行するスレッドを割り当てられたプロセッサでストア命令が実行された場合には、データ依存関係検出装置６０の対応するエントリに記憶された状態の読み出しを行う。より詳細には、ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、命令履歴テーブル６２の３つの入力ポートのうち対応するものに入力する。命令履歴テーブル６２は、入力されたエントリ番号に対応するエントリの内容を読み出し、論理和回路６１に出力する。同時に、自プロセッサがスレッド内の命令実行に関して投機実行状態にあるときに、自プロセッサがストア命令を実行した場合には、データ依存関係検出装置６０の対応するエントリに記憶された状態の読み出しを行う。より詳細には、ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、命令履歴テーブル６２の読み出し書き込み力ポートに入力する。命令履歴テーブル６２は、入力されたエントリ番号に対応するエントリの内容を読み出し、論理和回路６１に出力する。論理和回路６１は、命令履歴テーブル６２の１つの読み出し書き

込みポート及び３つの読み出しポートの出力の論理和を演算し、データ依存検出結果６４としてスレッド制御部に出力する。

【０２２０】

スレッド実行をデータ依存投機実行していた場合、他の３つのプロセッサのうち、自プロセッサが実行するスレッドよりも先行するスレッドを実行するプロセッサのいずれかが実行したストア命令の対象アドレスが、自プロセッサがそれまでデータ依存投機実行したロード命令の対象アドレスと等しいか、あるいは等しくない場合でも、エイリアスが発生して、命令履歴テーブル６２の同じエントリに割り当てられた場合、命令履歴テーブル６２からデータ依存投機実行されたロード命令が存在することを示す値が読み出されるため、スレッド間で正依存関係が存在する可能性があることが検出される。この場合、データ依存関係検出装置６０は、データ依存投機実行が失敗したことを示す値（例えば論理１）をデータ依存検出結果６４を通じてスレッド制御部に出力する。

【０２２１】

さらに、スレッド内の命令実行をデータ依存投機実行していた場合、自プロセッサが実行したストア命令の対象アドレスが、自プロセッサがそれまでデータ依存投機実行したロード命令の対象アドレスと等しいか、あるいは等しくない場合でも、エイリアスが発生して、命令履歴テーブル６２の同じエントリに割り当てられた場合、命令履歴テーブル６２からデータ依存投機実行されたロード命令が存在することを示す値が読み出されるため、スレッド内の命令間で正依存関係が存在する可能性があることが検出される。この場合も、データ依存関係検出装置６０は、データ依存投機実行が失敗したことを示す値（例えば論理１）をデータ依存検出結果６４を通じてスレッド制御部に出力する。

【０２２２】

スレッド制御部は、プロセッサ５９を含むデータ依存投機実行を行っていたプロセッサより、データ依存投機実行の失敗の通知を受け取ると、その失敗通知を出力したプロセッサ及び、その失敗通知を出力したプロセッサが実行していたスレッドよりも、プログラム順序で後に位置するスレッドを実行しているプロセッサに対して、データ依存投機実行の失敗による回復処理の実行要求を出力する。

データ依存投機実行の失敗による回復処理の実行要求の出力対象となった各プロセッサでは、データ依存投機実行の失敗による回復処理の実行要求が、回復処理実行要求４７を通じてプロセッサ制御部６３に通知される。

【０２２３】

スレッド単位のデータ依存投機実行の失敗だけではなく、スレッド内の命令単位のデータ依存投機実行の失敗によっても、データ依存投機実行を失敗したスレッドよりプログラム順序が後のスレッドを実行しているプロセッサに対して、データ依存投機実行の失敗による回復処理の実行を要求するのは、次に示す理由による。すなわち、スレッド内の命令のデータ依存投機実行が失敗したプロセッサは、正依存関係に違反してロード命令をデータ依存投機実行した可能性がある。そのため、前記データ依存投機実行が失敗したプロセッサが実行していた、データ依存投機実行に失敗したスレッドの実行結果は正しくない可能性がある。従って、前記データ依存投機実行に失敗したスレッドの実行結果を参照している可能性のある、失敗したスレッドよりも後続のスレッドを実行しているプロセッサに対しても、データ依存投機実行の失敗による回復処理が必要となる。

【０２２４】

スレッド単位あるいはスレッド内の命令単位のデータ依存投機実行の失敗により、データ依存投機実行の失敗による回復処理の実行要求の対象になったプロセッサでは、データ依存投機実行の失敗による回復処理の実行要求が、各プロセッサの回復処理実行要求４７を通じてプロセッサ制御部６３に通知される。その場合、上記各プロセッサは、データ依存投機実行を行ったスレッドの再実行を行うなど、データ依存投機実行の失敗による回復処理を実行することで、スレッドの実行結果の正しさを保証する。

【０２２５】

上述したように、この実施の形態によるデータ依存関係検出装置６０は、自プロセッサが実行するスレッドよりプログラム順序で先行するスレッドに含まれるストア命令から、自プロセッサが実行するスレッドに含まれるロード命令への正依存関係を検出する機能に加えて、自プロセッサが実行するスレッド内の命令間の正依存関係、すなわち、スレッド内のストア命令からロード命令への正依存関

係を検出する機能を有する。この２つの機能は、データ依存関係検出装置６０が備える命令履歴テーブル６２が、自プロセッサがロード命令をデータ依存投機実行する際に書き込みを行い、自プロセッサがストア命令をデータ依存投機実行する際に読み出しを行う読み出し書き込みポートを備えると共に、他のプロセッサがストア命令を実行する際に読み出しを行う複数の読み出しポートを備えることで実現される。これら機能により、スレッド並列処理において、スレッド単位のデータ依存投機実行の実現を可能とすることで、スレッド単位の並列実行によるプログラム実行のさらなる高速化を可能とするのみならず、スレッド内を構成する命令単位のデータ依存投機実行の実現を可能とすることで、スレッド内の非プログラム順序実行による高速化をも可能とする、という大きな効果を得ることができる。

【０２２６】

上記したデータ依存関係検出装置６０では、スレッド制御部が出力するスレッド順序４０を参照して、自プロセッサよりも先行するスレッドを実行している他のプロセッサを特定し、それら先行するスレッドを実行するプロセッサがストア命令を実行するときのみ、命令履歴テーブル６２の読み出しを行い、次に論理和関数６１でそれら読み出し結果の論理和を演算することで、先行するスレッドから自プロセッサが実行するスレッドへの正依存関係の存在の検出を実現した。一方、他の実現方法として、自プロセッサよりも先行するしないにかかわらず、他のプロセッサがストア命令を実行した場合には、常に命令履歴テーブル６２の読み出しを行っても良い。この場合、論理和関数６１でそれら読み出し結果の論理和を演算する際に、スレッド制御部が出力するスレッド順序４０を参照して、自プロセッサよりも先行するスレッドを実行している他のプロセッサを特定し、それら先行するスレッドによる読み出した結果のみ論理和を演算することで、先行するスレッドから自プロセッサが実行するスレッドへの正依存関係の存在の検出を実現することができる。

【０２２７】

また、上記した本発明の第３の実施の形態は、スレッド並列処理におけるスレッド単位のデータ依存投機実行において、自プロセッサでロード命令が実行され

た場合には、該ロード命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリに、自プロセッサでロード命令が実行されたことを示す値を格納し、自プロセッサあるいは他のプロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリから、自プロセッサでロード命令が実行されたか否かを示す値を読み出すことで、自プロセッサあるいは他のプロセッサで実行されたストア命令から、自プロセッサで実行されたロード命令への正依存関係が存在する可能性を検出することを特徴とした。しかしながら、この実施の形態は、スレッド間及びスレッド内の命令間の正依存関係が存在する可能性の検出にとどまらず、スレッド間及びスレッド内の命令間の逆依存関係や出力依存関係が存在する可能性の検出に対しても、同様の構成で容易に適用することができる。

【０２２８】

例えば、自プロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリに、自プロセッサでストア命令が実行されたことを示す値を格納し、自プロセッサあるいは他のプロセッサでロード命令が実行された場合には、該ロード命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリから、自プロセッサでストア命令が実行されたか否かを示す値を読み出すことで、自プロセッサあるいは他のプロセッサで実行されたロード命令から、自プロセッサで実行されたストア命令への逆依存関係が存在する可能性を検出する機能を有するデータ依存関係検出装置が実現される。

【０２２９】

または、他のプロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番

号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリから、自プロセッサでストア命令が実行されたか否かを示す値を読み出し、自プロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリから、自プロセッサでストア命令が実行されたか否かを示す値を読み出すのに加えて、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリに、自プロセッサでストア命令が実行されたことを示す値を格納することで、自プロセッサあるいは他のプロセッサで実行されたストア命令から、自プロセッサで実行されたストア命令への出力依存関係が存在する可能性を検出する機能を有するデータ依存関係検出装置が実現される。

【０２３０】

さらに、自プロセッサあるいは他のプロセッサでロード命令が実行された場合には、該ロード命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリから、自プロセッサでストア命令が実行されたか否かを示す値を読み出し、自プロセッサあるいは他のプロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリから、自プロセッサでロード命令が実行されたか否かを示す値及び自プロセッサでストア命令が実行されたか否かを示す値を読み出し、また、自プロセッサでロード命令が実行された場合には、該ロード命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリに、自プロセッサでロード命令が実行されたことを示す値を格納し、自プロセッサでストア命令が実行された場合には、該ストア命令の対象アドレスをハッシュ関数回路３１により命令履歴テーブル６２のエントリ番号に変換し、前記変換されたエントリ番号に対応する命令履歴テーブル６２のエントリに、自プロセッサでストア命令が実行されたことを示す値を格納することで、自

プロセッサあるいは他プロセッサで実行されたストア命令から、自プロセッサで実行されたロード命令への正依存関係が存在する可能性、及び自プロセッサあるいは他プロセッサで実行されたロード命令から、自プロセッサで実行されたストア命令への逆依存関係が存在する可能性、及び自プロセッサあるいは他のプロセッサで実行されたストア命令から、自プロセッサで実行されたストア命令への出力依存関係が存在する可能性、のすべてを検出する機能を有するデータ依存関係検出装置が実現される。

【０２３１】

上記した各実施例において、データ依存関係検出装置を備えたプロセッサは、好ましくは半導体基板上に集積化される半導体集積回路（ＬＳＩ）装置として実現される。

【０２３２】

【発明の効果】

以上説明したように、本発明のデータ依存関係検出装置によれば、データ依存投機実行時において、ストア命令からロード命令への正依存関係の存在を厳密に検出するのではなく、ストア命令からロード命令への正依存関係の存在の可能性を検出することを特徴としている。任意のロード命令の対象アドレスは、ハッシュ関数回路により、命令履歴テーブルの特定のエントリに重複を許して割り当てるため、従来のデータ依存関係検出装置と異なり、実行されたロード命令の対象アドレス自体をすべて記憶しておく必要がなく、またストア命令の対象アドレスと比較するためのアドレス比較器も必要としない。すなわち、本発明によるデータ依存関係検出装置によれば、データ依存投機実行の成否判定に必要なハードウエア量が小さいという効果を得ることができる。さらに、データ依存投機実行することのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもデータ依存投機実行による実行性能の向上を享受することのできるという効果を得ることができる。また、本発明のデータ依存関係検出装置によれば、ストア命令の実行時の正依存関係の検出処理において、ストア命令の対象アドレスとロード命令の対象アドレスの比較を必要としない。さらに、命令履歴テーブルの読み出すエントリは１つだけなので、

すべてのエントリ出力の論理和を演算する必要もない。すなわち、正依存関係の検出の処理時間が短いため、プロセッサの動作周波数を高速化することが容易であるという効果を得ることができる。

【０２３３】

また、本発明のデータ依存関係検出装置によれば、スレッド並列処理におけるスレッド単位のデータ依存投機実行においても、データ依存投機実行の成否判定に必要なハードウエア量が小さいという効果を得ることができる。さらに、データ依存投機実行を行うことのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもスレッド単位のデータ依存投機実行による実行性能の向上を享受することのできるという効果を得ることができる。さらに、スレッド間の依存関係の検出の処理時間が短く、プロセッサの動作周波数を高速化することが容易であるという効果を得ることができる。

【０２３４】

さらに、本発明のデータ依存関係検出装置によれば、スレッド並列処理におけるスレッド単位のデータ依存投機実行及び、スレッド内の命令単位のデータ依存投機実行を可能とし、さらにそれらデータ依存投機実行の成否判定に必要なハードウエア量が小さいという効果を得ることができる。さらに、データ依存投機実行することのできるロード命令の数が、データ依存関係検出装置のハードウエア量に制限されないため、少ないハードウエア量でもスレッド単位のデータ依存投機実行及びスレッド内の命令単位のデータ依存投機実行による実行性能の向上を享受することのできるという効果を得ることができる。さらに、データ依存関係の検出の処理時間が短く、プロセッサの動作周波数を高速化することが容易であるという効果を得ることができる。

【図面の簡単な説明】

【図１】

本発明の第１の実施の形態によるデータ依存関係検出装置を含むプロセッサの構成を示すブロック図である。

【図２】

本発明の第１の実施の形態によるデータ依存関係検出装置の動作を示すタイミング図である。

【図３】

本発明の第１の実施の形態によるデータ依存関係検出装置の構成を示すブロック図である。

【図４】

本発明の第１の実施の形態によるデータ依存関係検出装置の動作の具体例を説明する説明図である。

【図５】

本発明のデータ依存関係検出装置が備えるハッシュ関数回路の実施例を示す図である。

【図６】

本発明の第２の実施の形態によるデータ依存関係検出装置を含むマルチプロセッサの構成を示すブロック図である。

【図７】

本発明の第２の実施の形態によるデータ依存関係検出装置の動作の具体例を説明する説明図である。

【図８】

本発明の第３の実施の形態によるデータ依存関係検出装置の構成を示すブロック図である。

【図９】

データ依存投機実行の動作を説明する説明図である。

【図１０】

従来のデータ依存関係検出装置の構成を示すブロック図である。

【符号の説明】

１０　データ依存関係検出装置

１１　ハッシュ関数回路

１２　命令履歴テーブル

１３　投機実行フラグ

１４　実行命令

１５　ロード・ストアアドレス

１６　データ依存検出結果

１７　プロセッサ制御部

１８　命令実行部

２０　データ依存関係検出装置

２１　ハッシュ関数回路

２２　命令履歴テーブル

３０　データ依存関係検出装置

３１　ハッシュ関数回路

３２　命令履歴テーブル

３３　投機実行フラグ

３４　実行命令

３５　ロード・ストアアドレス

３６　データ依存検出結果

３７　ロード・ストアアドレス

３８　実行命令

３９　投機実行フラグ

４０　スレッド順序

４１　論理和回路

４５　ロード・ストアアドレス

４６　実行命令

５０、５１、５２、５３　プロセッサ

５５　スレッド制御部

５７　プロセッサ制御部

６０　データ依存関係検出装置

６１　論理和回路

６２　命令履歴テーブル

６３　プロセッサ制御部

６４　データ依存検出結果

１００　データ依存関係検出装置

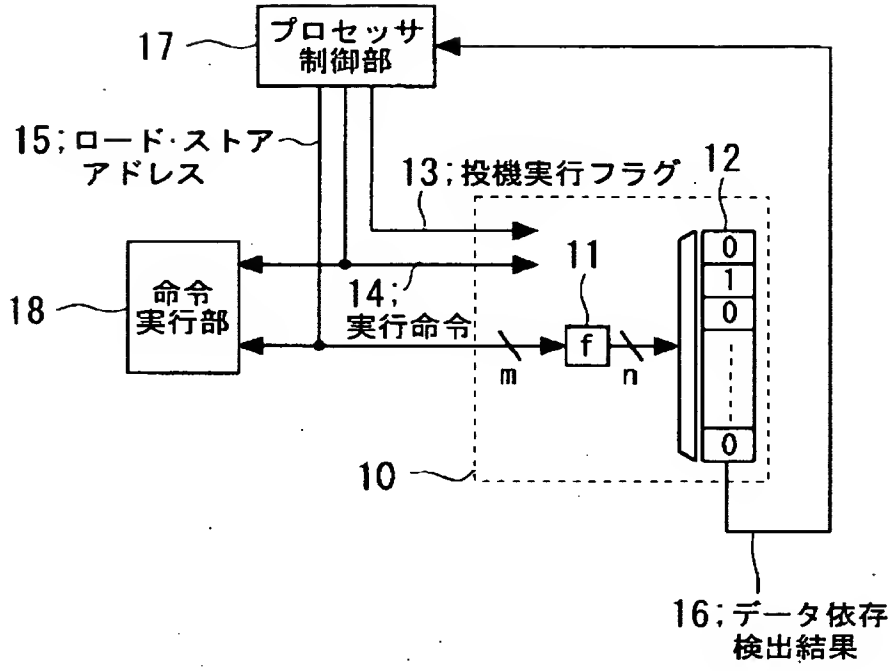
１０１　アドレスバッファ

１０２　アドレス比較器

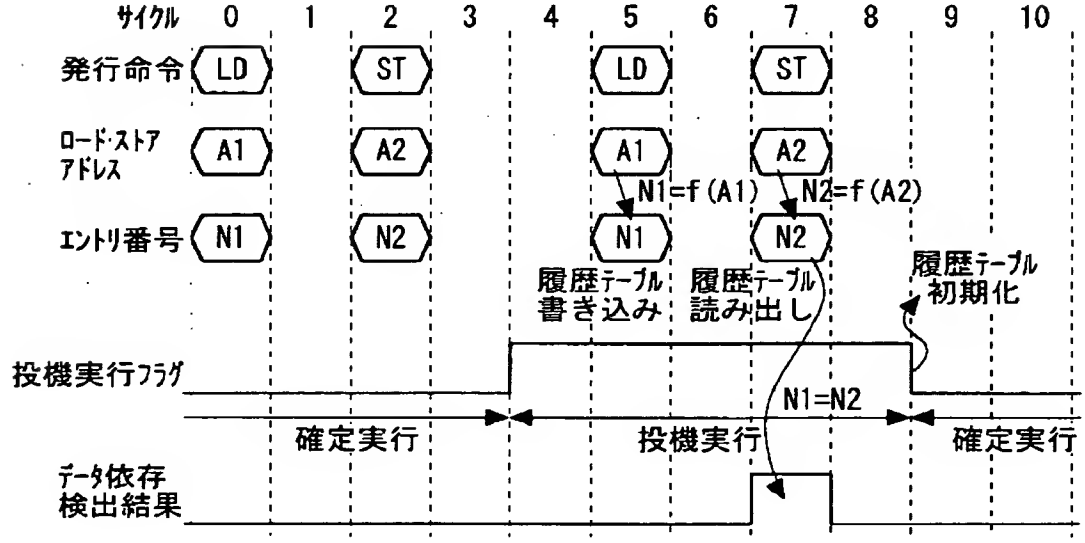
１０３　論理和回路

【書類名】　　図面

【図１】

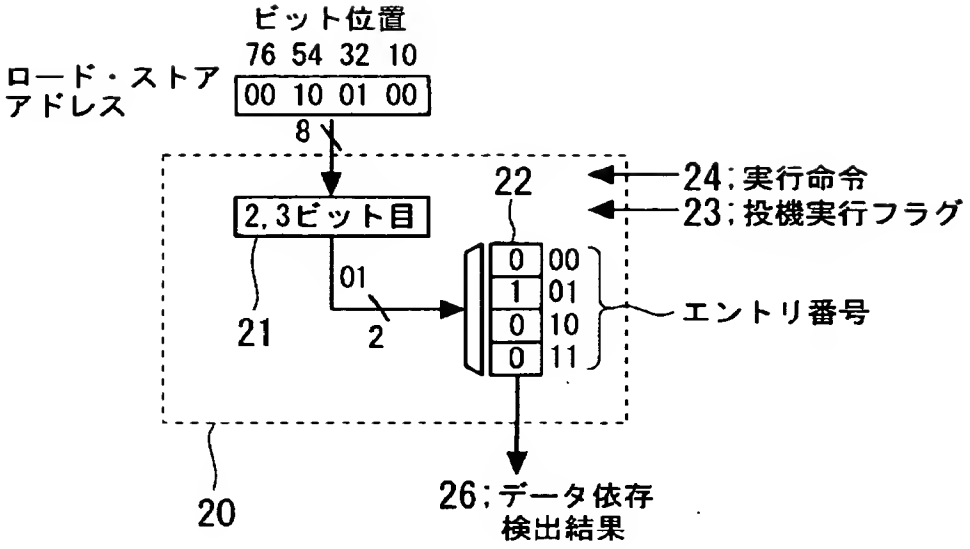


【図２】

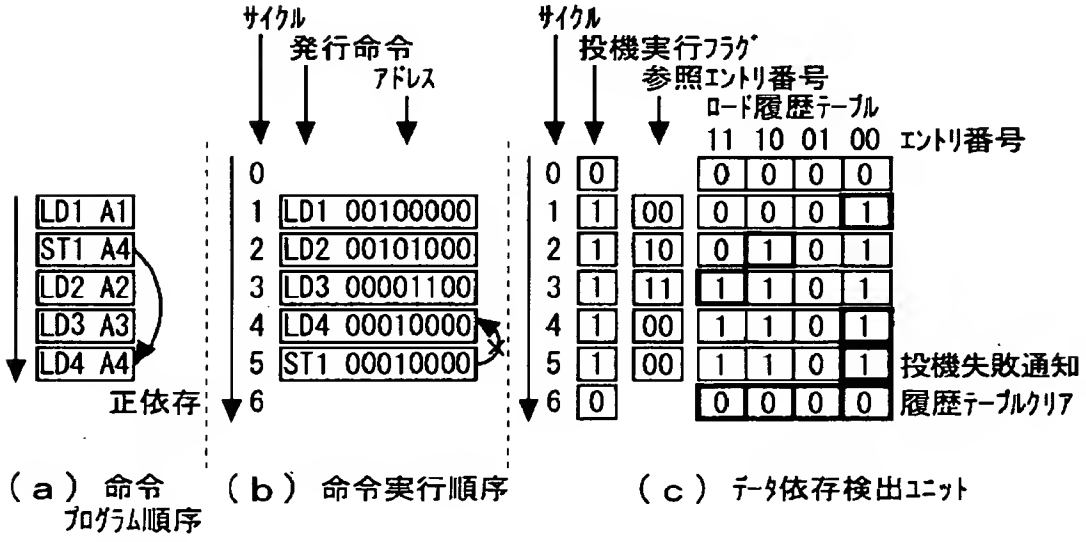


出証特２００１－３０７７８１３

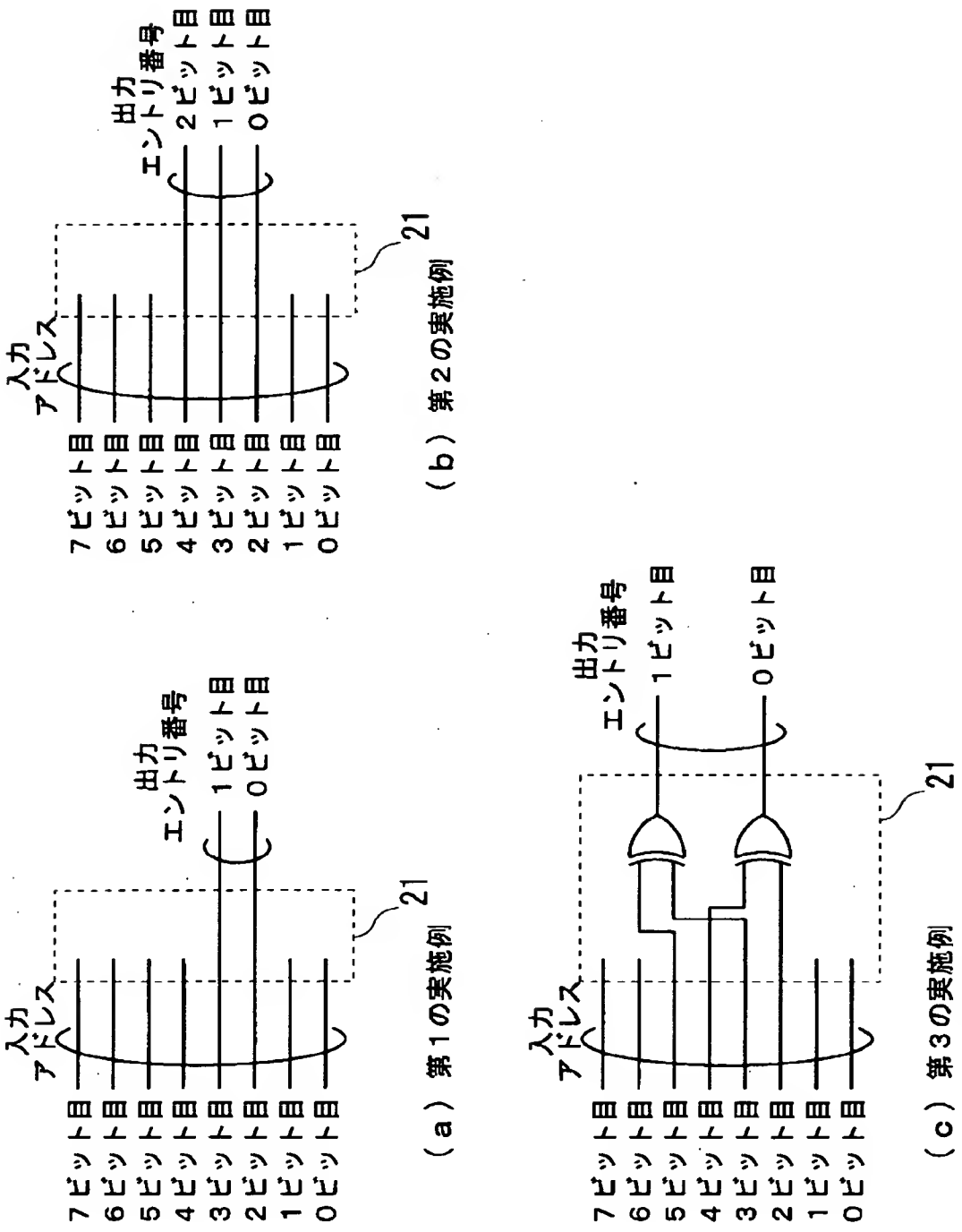
【図３】


ビット位置
76 54 32 10
ロード・ストア
アドレス
00 10 01 00
8
2,3ビット目
01
21
2
22
0 00
1 01
0 10
0 11
エントリ番号
24；実行命令
23；投機実行フラグ
20
26；データ依存
検出結果

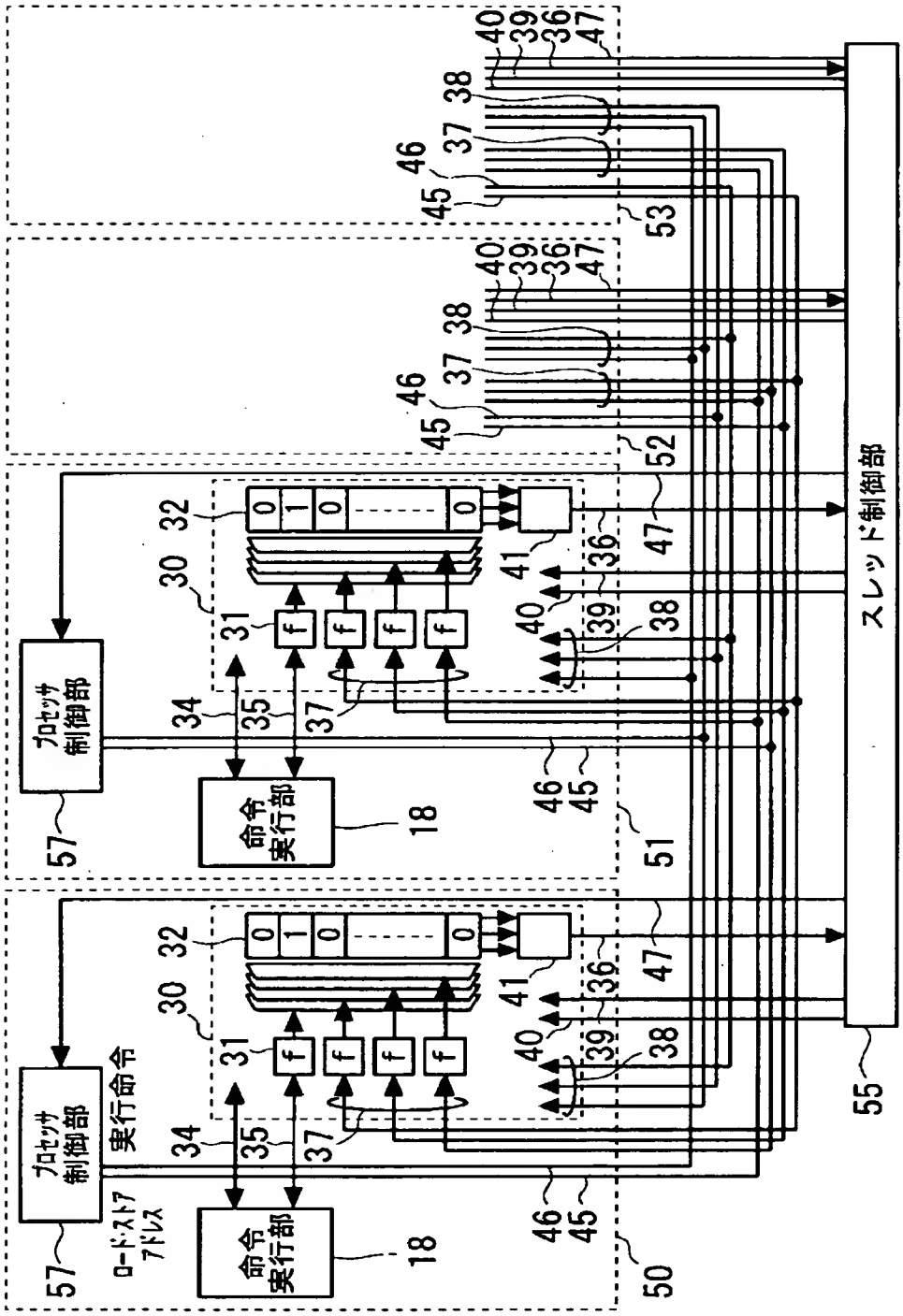
【図４】


サイクル
発行命令
アドレス
LD1 A1
ST1 A4
LD2 A2
LD3 A3
LD4 A4
正依存
0
1 LD1 00100000
2 LD2 00101000
3 LD3 00001100
4 LD4 00010000
5 ST1 00010000
6
サイクル
投機実行フラグ
参照エントリ番号
ロード履歴テーブル
11 10 01 00 エントリ番号
0 0 0 0 0
1 1 00 0 0 0 1
2 1 10 0 1 0 1
3 1 11 1 1 0 1
4 1 00 1 1 0 1
5 1 00 1 1 0 1 投機失敗通知
6 0 0 0 0 0 履歴テーブルクリア
（ａ）命令
プログラム順序
（ｂ）命令実行順序
（ｃ）データ依存検出ユニット

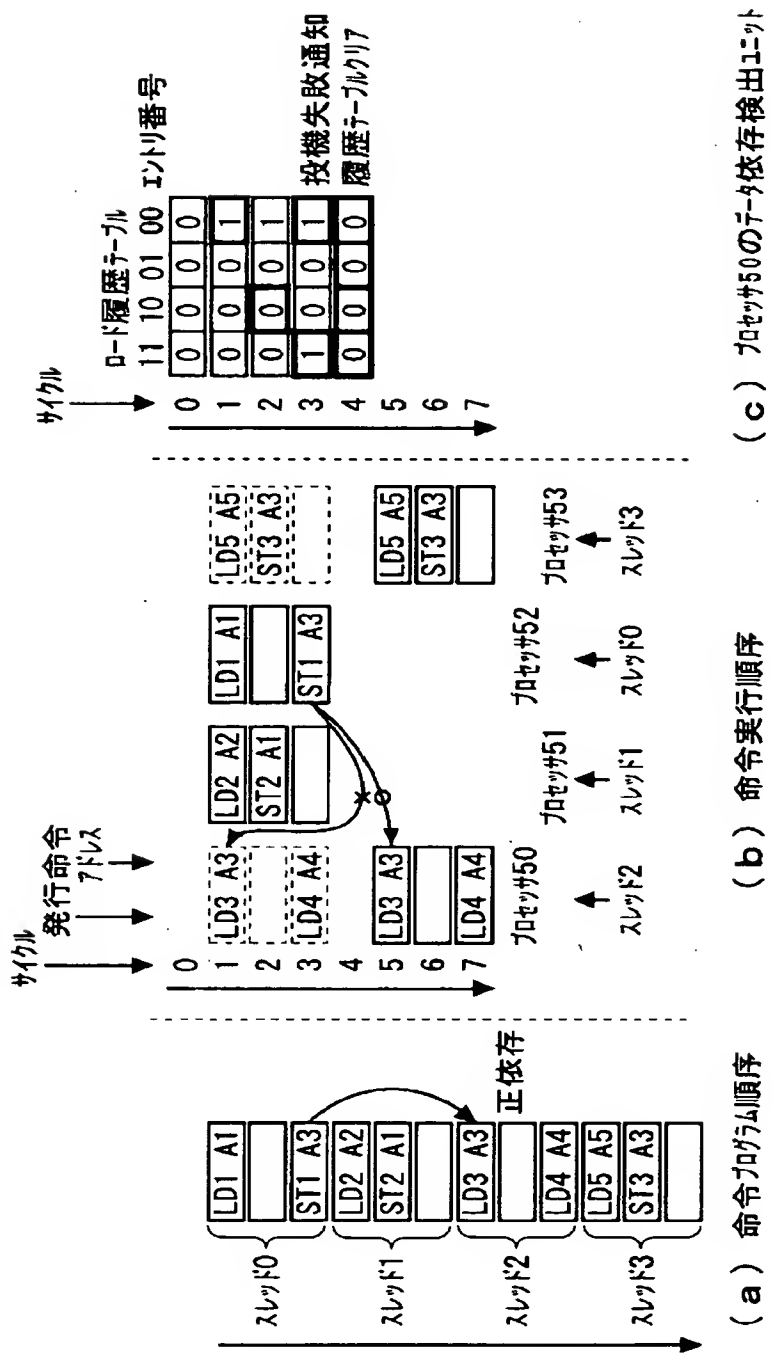
【図５】


入力
アドレス
出力
エントリ番号
7ビット目
6ビット目
5ビット目
4ビット目
3ビット目
2ビット目
1ビット目
0ビット目
21
（ａ）第１の実施例
（ｂ）第２の実施例
（ｃ）第３の実施例

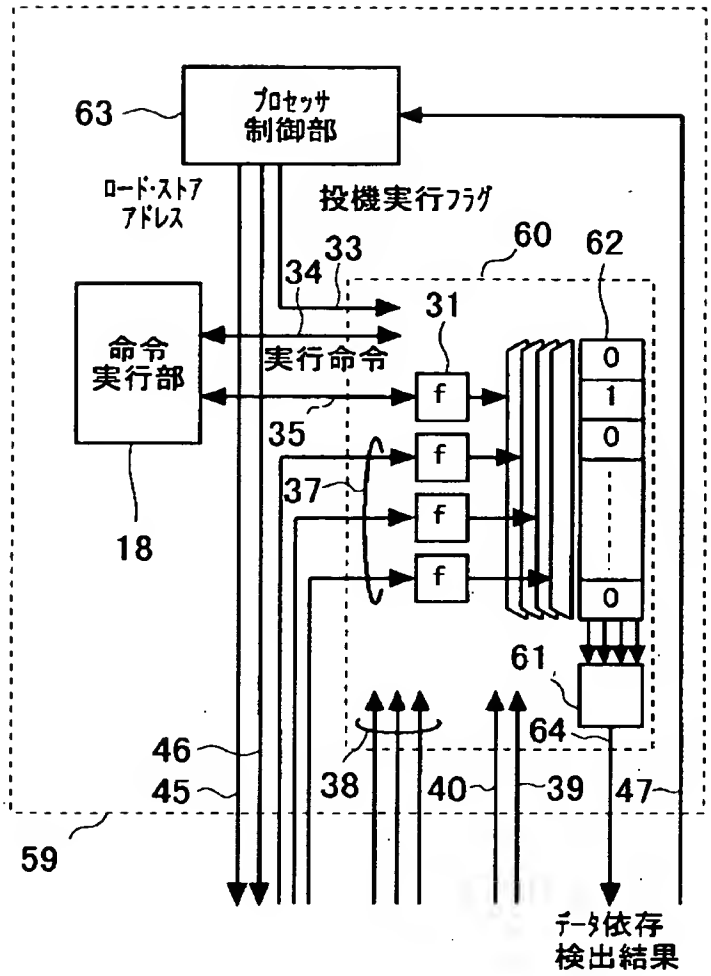
【図６】



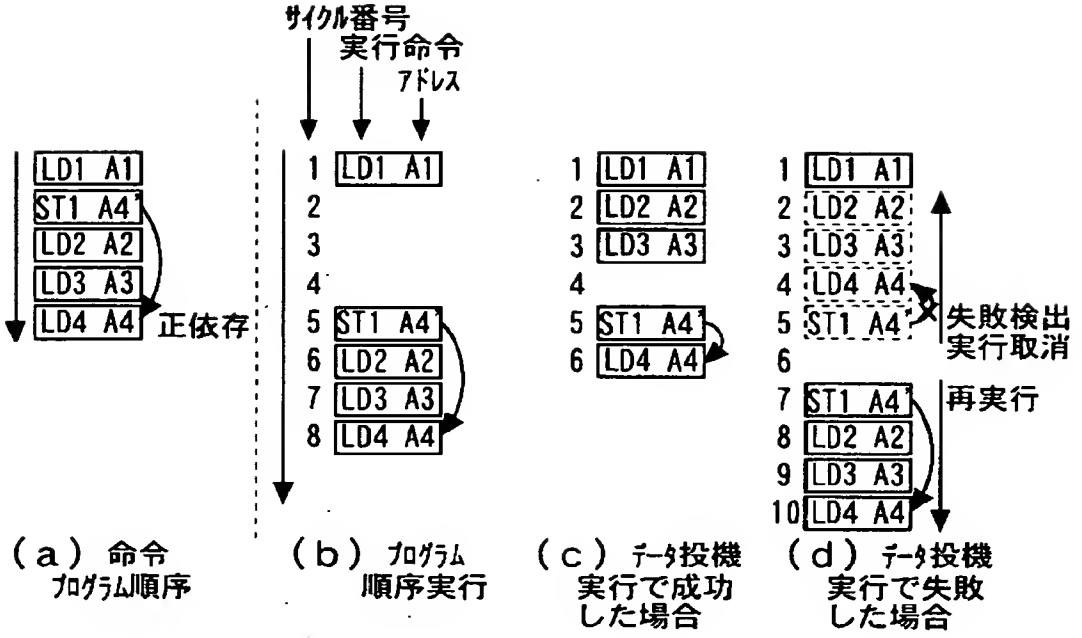
【図７】


サイクル
発行命令
アドレス
LD3 A3
LD4 A4
LD2 A2
ST2 A1
LD1 A1
ST1 A3
LD5 A5
ST3 A3
プロセッサ50
プロセッサ51
プロセッサ52
プロセッサ53
スレッド2
スレッド1
スレッド0
スレッド3
（ｂ）命令実行順序
ロード履歴テーブル
エントリ番号
11 10 01 00
投機失敗通知
履歴テーブルクリア
（ｃ）プロセッサ50のデータ依存検出ユニット
正依存
（ａ）命令プログラム順序

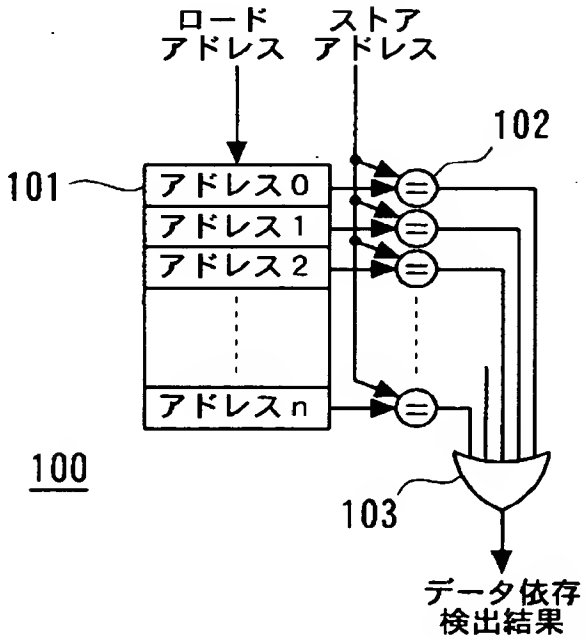
【図８】


63
プロセッサ
制御部
ロード・ストア
アドレス
投機実行フラグ
33
34
60
62
31
命令
実行部
実行命令
0
1
0
0
f
f
f
f
35
37
18
61
64
46
45
38
40
39
47
59
データ依存
検出結果

【図９】


サイクル番号
実行命令
アドレス
LD1 A1
ST1 A4
LD2 A2
LD3 A3
LD4 A4
正依存
1 LD1 A1
2
3
4
5 ST1 A4
6 LD2 A2
7 LD3 A3
8 LD4 A4
1 LD1 A1
2 LD2 A2
3 LD3 A3
4
5 ST1 A4
6 LD4 A4
1 LD1 A1
2 LD2 A2
3 LD3 A3
4 LD4 A4
5 ST1 A4
失敗検出
実行取消
6
7 ST1 A4
再実行
8 LD2 A2
9 LD3 A3
10 LD4 A4
（ａ）命令
プログラム順序
（ｂ）プログラム
順序実行
（ｃ）データ投機
実行で成功
した場合
（ｄ）データ投機
実行で失敗
した場合

【図１０】


ロード
アドレス
ストア
アドレス
102
101
アドレス０
アドレス１
アドレス２
アドレスｎ
100
103
データ依存
検出結果

【書類名】　　要約書

【要約】

【課題】

データ依存投機実行することのできるロード命令の数が、ハードウエア量に制限されず、小さいハードウエア量でもデータ依存投機実行による実行性能の向上を享受することができ、依存検出の処理時間が短く、プロセッサの動作周波数を高速化することが容易である、データ依存関係検出装置の提供。

【解決手段】

プロセッサが非プログラム順序でロード命令またはストア命令の実行を行う際に、前記ロード命令またはストア命令が処理対象とするアドレス間の依存関係の存在の可能性を検出するデータ依存関係検出装置であって、実際に前記依存関係が存在する場合は必ず依存関係が存在する旨を検出するが、実際に前記依存関係が存在しない場合でも依存関係が存在する旨を検出する場合があることを許容することを特徴とし、ロード命令及びストア命令が処理対象とするアドレスを一意な番号に変換するアドレス変換手段と、前記変換された番号に対応して、ロード命令又はストア命令が実行されたか否かを記憶する複数の記憶手段から構成される実行履歴記憶手段とを備える。

【選択図】

図１

出　願　人　履　歴　情　報

識別番号　　〔０００００４２３７〕

１．変更年月日　　１９９０年　８月２９日
［変更理由］　新規登録
住　所　東京都港区芝五丁目７番１号
氏　名　日本電気株式会社